digelec

MODEL 804
UNIVERSAL EPROM/EEPROM
PROGRAMMER AND SIMULATOR
USER MANUAL

Publication Number MN-100/0386/RE

DIGELEC INC. (U.S.A.): 1602 LAWRENCE AVENUE, OCEAN, NEW JERSEY 07712. TEL: (201)493-2420, TLX: 754-098
DIGELEC AG. (EUROPE): DÖRFLISTRASSE 14, CH-8057 ZÜRICH, SWITZERLAND. TEL (01) 312 46 22, TLX: 823 795 DIGE CH

COPYRIGHT

(C) DIGELEC INC. 1986

This document contains proprietary information that is protected by copyright. No part of this document may be reproduced without the written consent of the publisher.

The information furnished by DIGELEC INC. in this publication is believed to be accurate and reliable. However, no responsibility is assumed by DIGELEC INC. for its use, nor for any consequential or other damages to the equipment that it accompanies resulting from its use.

DIGELEC INC. reserves the right to revise this publication and make changes in its content without notification, at any time.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION....................................1

CHAPTER 2 PRODUCT SPECIFICATIONS........................2-1

2.1 FUNCTIONAL SPECIFICATIONS.....................2-1

2.2 DATA TRANSLATION FORMATS......................2-1

2.3 POWER REQUIREMENTS...........................2-1

2.4 PHYSICAL SPECIFICATIONS.......................2-2

2.5 ENVIRONMENTAL SPECIFICATIONS..................2-2

2.6 OPTIONS AND ORDERING INFORMATION..............2-2

CHAPTER 3 GETTING ACQUAINTED...........................3-1

3.1 AN OVERVIEW OF THE FRONT PANEL................3-1

3.2 KEYS ON THE FRONT PANEL.......................3-2

3.2.1 POWER ON/OFF.............................3-3
3.2.2 MODE KEYS................................3-3
3.2.3 HEXADECIMAL KEYBOARD.....................3-4
3.2.4 CONTROL KEYS.............................3-4

3.3 OPERATING INDICATORS..........................3-4

3.3.1 OPERATING STATE INDICATORS...............3-5
3.3.2 THE FUNCTION LEDS........................3-5

3.4 RESPONSES.....................................3-6

3.4.1 LED INDICATORS...........................3-6
3.4.2 AUDIBLE SIGNAL...........................3-7

3.5 INPUT/OUTPUT..................................3-7

3.5.1 DISPLAY..................................3-7
3.5.2 RS-232C..................................3-7
3.5.3 SIM CONNECTOR............................3-7

CHAPTER 4 GETTING STARTED..............................4-1

4.1 SYSTEM TEST..................................4-2

4.2 FUNCTIONS WITHOUT INITIALIZATION.............4-3

4.3 SELECTING THE DEVICE.........................4-4

4.3.1 AUTOMATIC IDENTIFIER......................4-4
4.3.2 ENTERING THE DEVICE CODE..................4-5
4.3.3 SCROLLING TO LOCATE THE DEVICE............4-5

4.4 DEVICE ADAPTOR...............................4-6

4.5 INITIALIZING PARAMETERS FOR TRANSMISSION.....4-7

CHAPTER 5 KEY MODE....................................5-1

5.1 THE FUNCTIONS................................5-1

5.2 SELECTING A FUNCTION.........................5-2

5.3 LOADING DATA INTO RAM........................5-2

5.3.1 IN: KEYBOARD..............................5-2
5.3.2 IN: EPROM.................................5-4
5.3.3 IN: RS-232C...............................5-6

5.4 FUNCTIONS PERFORMED IN RAM...................5-8

5.4.1 SPECIFYING THE DATA BLOCK.................5-8
5.4.2 FILL CONSTANT.............................5-10
5.4.3 MOVE......................................5-11
5.4.4 SEARCH....................................5-13
5.4.5 SPLIT/SHUFFLE.............................5-14

5.5 VIEWING THE CONTENTS OF RAM..................5-17

5.5.1 OUT: DISPLAY..............................5-17
5.5.2 OUT: RS-232C..............................5-18

5.6 EPROM FUNCTIONS..............................5-20

5.6.1 CHECK.....................................5-20

5.6.1.1 BLANK CHECK............................5-22
5.6.1.2 SUMCHECK...............................5-23
5.6.1.3 ILLEGAL BIT TEST.......................5-24

5.6.2 VERIFY....................................5-26
5.6.3 PROGRAM...................................5-29
5.6.4 ERASE.....................................5-32

5.7 VIEWING THE DATA FROM THE EPROM..............5-33

5.7.1 OUT: DISPLAY..............................5-33
5.7.2 OUT: RS-232C..............................5-34

CHAPTER 6 HANDLING THE EPROM..................................6-1

6.1 SELECTING AN EPROM...........................6-1

6.2 INSERTING THE EPROM..........................6-1

6.3 PRECAUTIONS..................................6-1

CHAPTER 7 REMOTE MODE..................................7-1

7.1 INTRODUCTION.................................7-1

7.2 SET-UP.......................................7-1

7.3 OPERATION....................................7-2a

7.4 FUNCTIONS WITH PARAMETERS....................7-2b

7.5 SELECTING THE EPROM TYPE.....................7-2b

7.6 SETTING PARAMETERS FOR TRANSMISSION..........7-3

7.7 DATA TRANSFER COMMANDS.......................7-4

7.8 RAM FUNCTIONS................................7-4

7.9 EPROM FUNCTIONS..............................7-5

7.10 ERRORS.......................................7-7

7.11 ERROR HANDLING...............................7-8

7.11.1 ERROR STATUS WORD.........................7-8
7.11.2 ERROR CODES...............................7-9

CHAPTER 8 SERIAL PORT....................................8-1

8.1 INTRODUCTION....................................8-1

8.2 CONNECTION DETAILS...........................8-1

8.3 INITIALIZATION................................8-3

8.3.1 DATA FORMAT................................8-3
8.3.2 BAUD RATE..................................8-3
8.3.3 NUMBER OF STOPBITS.........................8-3
8.3.4 FRAME......................................8-3
8.3.5 PARITY BIT.................................8-3
8.3.6 ECHO MODE..................................8-4
8.3.7 ADDRESS ERROR HANDLING.....................8-4
8.3.8 ADDRESS DISPLAY IN RS-232C FUNCTIONS......8-5

8.4 ERROR HANDLING................................8-5

8.4.1 TRANSMISSION AND HARDWARE FAULTS..........8-5
8.4.2 CHARACTER ERROR............................8-5
8.4.3 CHECKSUM AND ADDRESS ERROR.................8-5

8.5 DATA FORMATS..................................8-6

8.5.1 MANUAL.....................................8-6
8.5.2 HEX-SPACE..................................8-7
8.5.3 ASCII-BNPF.................................8-7
8.5.4 INTEL-HEX..................................8-8
8.5.5 MOTOROLA EXORCISER.........................8-10
8.5.6 TEKTRONIX-MILLENIUM........................8-11
8.5.7 MOS-TECH...................................8-14

CHAPER 9 EPROM SIMULATION.............................9-1

9.1 INTRODUCTION..................................9-1

9.2 TECHNICAL DATA................................9-1

9.3 TIMING CONSIDERATIONS.........................9-2

9.4 CONNECTION DETAILS............................9-3

9.5 OPERATION.....................................9-3

Appendix A A List Of Programmable Devices....................A-1
Appendix B Upgrading the Software of the Model-804.............A-9
Appendix C Interfacing the Model-804 with the INTEL MDS........A-11
Appendix D Interfacing the Model-804 with the PMDS............A-18

FIGURES

1. SYSTEM CONFIGURATION..1-1

2. THE FRONT PANEL...3-1

3. SET-UP PROCEDURE..4-1

4. SELECTING A PARAMETER.......................................4-8

5. THE FUNCTIONS...5-1

6. CONNECTION TO THE SERIAL PORT...............................8-2

7. TIMING DIAGRAM..9-2

8. SIM CONNECTOR...9-3

TABLES

1. SYSTEM FAULT CONDITIONS.....................................4-3

2. INITIALIZING THE PARAMETERS FOR TRANSMISSION...............4-9

CHAPTER 1

INTRODUCTION

The Model-804 is a portable microprocessor based E/EPROM programmer and simulator, used as an engineering tool for laboratory and field services. The system is capable of programming a wide variety of MOS EPROMs, EEPROMs, and single chip microprocessors (see Appendix A for a list of programmable devices).

The Model-804 is a flexible system. It's programming software is stored in EPROMs which can be replaced or modified to accommodate future components. It can be used as either a stand-alone keyboard operated unit, or it can be interfaced with another system via the RS-232C serial port. The Model-804 can also be operated remotely by an external computer using a variety of translation formats.

The Model-804 is user friendly. It communicates with its operator interactively, through a built-in hexadecimal keyboard and 16-character alphanumeric display. The operator is prompted, at each stage of the dialogue, to enter the required data, and entries are immediately acknowledged by the programmer with a series of responses. A wide variety of editing functions are accessible for manipulating and preparing the desired programmed data. The user RAM buffer of the Model-804 can be expanded to 64K bytes.

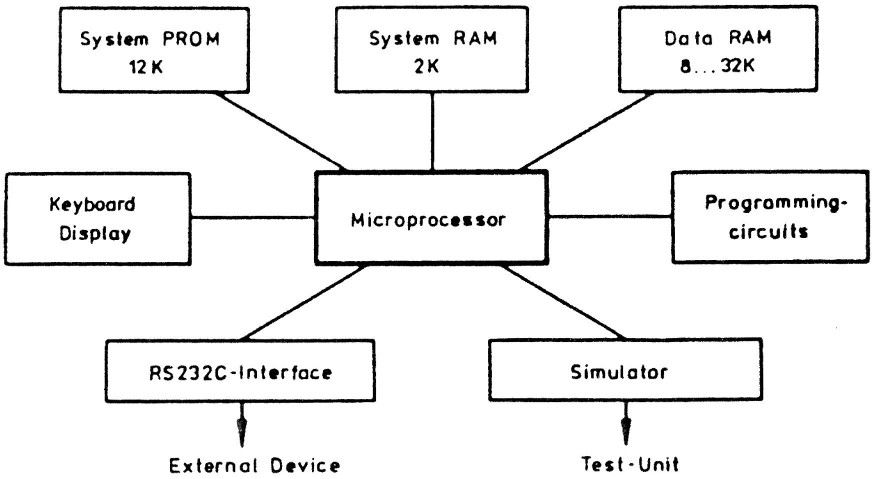


FIGURE 1 SYSTEM CONFIGURATION

How To Use This Manual

This manual is designed for both the first time user and the experienced user. A first time user will want to take some time to read through this manual carefully, before beginning to work with the Model-804. An experienced programmer will find this manual a handy reference book.

The Structure Of This Manual

Chapter 1 is an introduction to the Model-804.

Chapter 2 contains the product specifications, and information on optional equipment available for the Model-804.

In Chapter 3 you will become acquainted with the location and the function of the various features of the Model-804 front panel.

In Chapter 4 the preliminary steps for using your programmer are reviewed. This chapter discusses the internal system test that takes place when you get started, and the initialization procedure that will precede all data transfers to an EPROM.

Chapter 5 is a discussion of the functions in KEY mode, in which the Model-804 is operated manually, by its keys. The functions are the tools for receiving and transmitting data to and from RAM, and manipulating data in either RAM or the EPROM. In chapter 5 these functions will be discussed individually, and examples will be provided for each function.

Chapter 6 contains important information on handling your EPROM. You will want to know how to insert it, and what precautions to take so as not to damage it.

Chapter 7 is about operating the Model-804 from an external terminal or computer, in the REM mode. This chapter details the general procedure, as well as the specific codes used to command the Model-804 functions from a remote source.

Chapter 8 contains further details on the RS-232 serial port, including connection details, initialization at the external computer, error handling, and a detailed discussion of data formats.

Chapter 9 is about using your Model-804 as a simulator for testing the program before it is actually entered into the EPROM.

Appendix A contains a list of the programmable devices supported by the Model-804.

Appendix B contains the instructions for upgrading your Model-804.

Appendix C contains information about interfacing the Model-804 to the INTEL MDS.

Appendix D contains information about interfacing the Model-804 to the PMDS.

CHAPTER 2

PRODUCT SPECIFICATIONS

2.1 FUNCTIONAL SPECIFICATIONS

General Architecture:	Microprocessor-controlled (Z-80)
Sockets:	One 28-pin Textool socket
Data RAM:	Standard 16k byte, expandable to 64k bytes
I/O port:	Standard RS-232C Serial Port
Remote control:	Standard
Baud Rates:	50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200.
Keyboard:	16 Hex keys and 8 function keys
Display:	16-character alphanumeric
Audible signal:	For keying and operating errors.
Simulator:	Standard EPROM simulator with 300 ns access time.

2.2 DATA TRANSLATION FORMATS

Manual

Hex-Space

BNPF

Intel-Hex

Motorola

Tektronix

Mos-Tech

2.3 POWER REQUIREMENTS

Operating Voltages:	100-110V / 200-240V Switch Selectable
Frequency:	50/60 Hz

2.4 PHYSICAL SPECIFICATIONS

Width:	32.0 cm (12.6 in.)
Height:	8.5 cm (33 in.)
Depth:	20.8 cm (8.2 in.)
Weight:	2.3 Kg (5 lb.)

2.5 ENVIRONMENTAL SPECIFICATIONS

Operating Temperature Range:	0 to 40 degrees C
Relative Humidity:	85%

2.6 OPTIONS AND ORDERING INFORMATION

Part no.	Description
804-70	RAM expansion kit (not including RAM devices)
804-72	RAM expansion to 32k bytes (including 804-70)
8041	Device adapter for EEPROM 2815 and 2816.
8042	Device adapter for 8755A
8043	Device adapter for the single-chip microprocessor 8751, 8751H and 8761H
8044	Device adapter for EEPROM 2817
8045	Device adapter for single-chip microprocessor 874x.
8046	Device adapter for 5v EEPROMs.
804-SW	Software update kit for the Model-804

Simulator cables for:

804-2020	2716, 27C16, 2732, 2732A, 27C32, 2758, HN 48016.
804-2021	2764, 27C64, 27128, 27256, 27512.
804-2024	TMS2704, TMS2708.
804-2025	TMS2716.
805-2026	TMS2532, MCM68732, MCM68764, MCM68766.
804-2028	TMS2564.

Interface cables for:

804-2030	Philips, PMDS.
804-2032	Intel, MDS (please specify system and port number).

Note: Specifications are subject to change without notice.

CHAPTER 3

GETTING ACQUAINTED

3.1 AN OVERVIEW OF THE FRONT PANEL

In this chapter you will get acquainted with the front panel. You will learn where the keys are located, the function of these keys and some of the other features of the Model-804.

Figure-1 is a diagram of the front panel. We will be refering to this diagram throughout the chapter and you will find it useful to return to this figure as the items on the front panel are discussed.

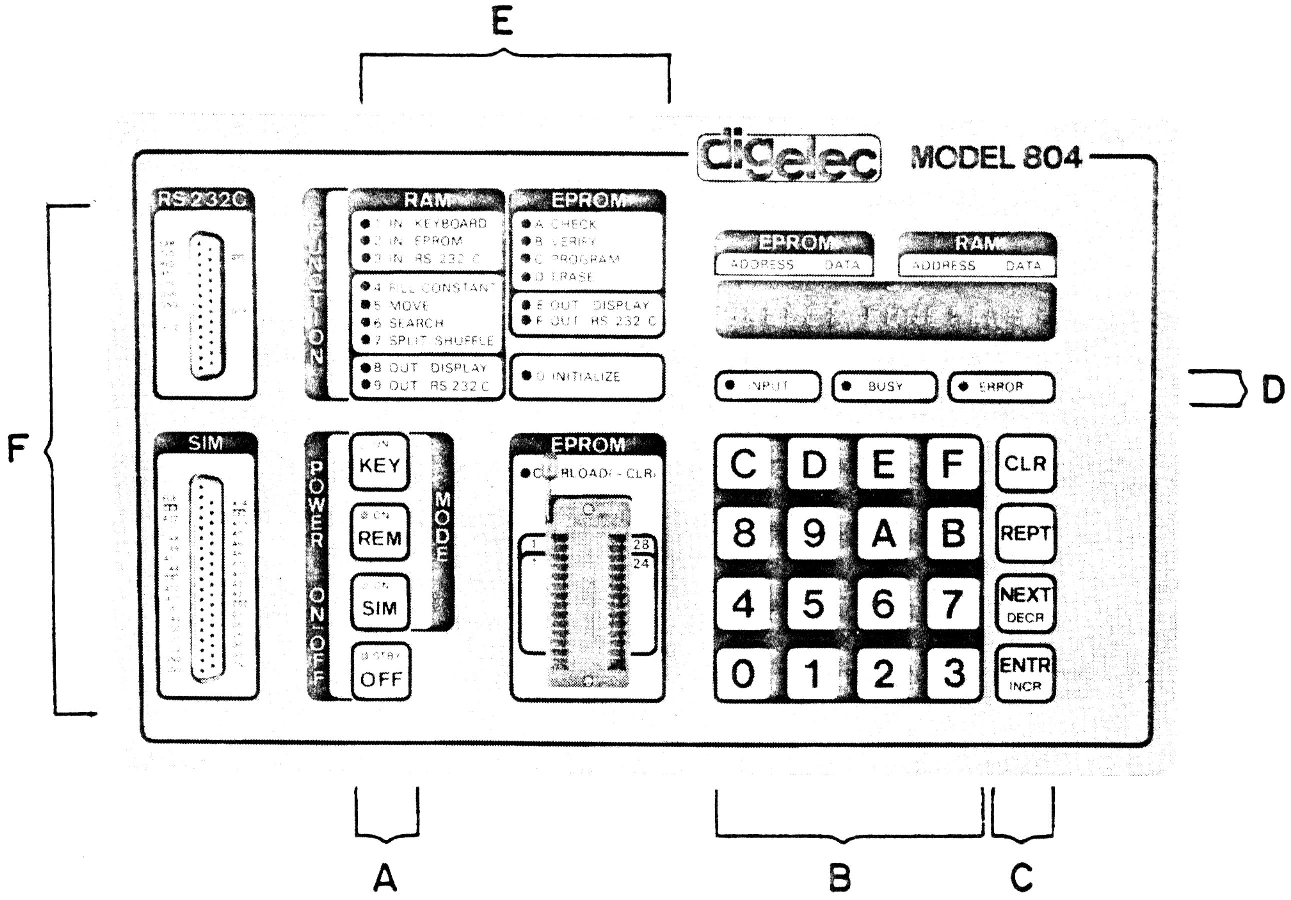


FIGURE 2 THE FRONT PANEL

In the middle of the front panel you will find the socket for plugging in the EPROM. Starting with the keys at the left of the EPROM socket, and moving counterclockwise, locate the following features on the front panel:

Group A:	At the left of the EPROM socket is the POWER ON/OFF and MODE keys. These keys turn the power on and let the Model-804 know how it will be operated.
Group B:	The HEXADECIMAL KEYBOARD, located to the right of the EPROM socket, is used to choose functions and to enter data directly.
Group C:	The CONTROL keys, located at the right of the hexadecimal keyboard, are used to respond to prompts from the Model-804. They are labeled CLR, REPT, NEXT/DECR and ENTR/INCR.
Group D:	The INDICATOR LEDs (Light Emitting Diodes), located above the hexadecimal keyboard, specify the operating condition of the Model-804. They are labeled INPUT, BUSY and ERROR.
Group E:	The FUNCTION LEDs are presented in two columns under the heading RAM and EPROM. The LEDs indicate the function that is in progress.
Group F:	The RS-232 socket and the SIM socket are used to interface with an external instrument for data input and output and for simulating EPROM functions.

3.2 KEYS ON THE FRONT PANEL

This section will describe the function of each key on the front panel. Items will be described sequentially in the order that they appear in Figure-1. It will be useful for you to refer to the diagram throughout this section.

Groups A, B and C are command keys used to control the operation of the Model-804. Use these keys to turn on your programmer, choose the mode of operation, input data and respond to prompts.

3.2.1 POWER ON/OFF (Group A)

The Model-804 has two power-down states.

1. power not connected
2. the STANDBY state

STANDBY is used for taking the Model-804 out of operation without disconnecting the power. When in the STANDBY state, the LED corresponding to the OFF-Key will be lit. Power consumption will be reduced to 2 watts, and the data and system RAM will retain their contents. Stored data and initialization parameters will be saved.

If a CMOS device is selected, the message REMOVE DEVICE will appear when pressing the 'OFF' key. Power down by pressing 'ENTR'.

3.2.2 MODE KEYS

The MODE-keys are used to bring the Model-804 up to the operating state and to select the basic operating mode. Operating a MODE-key causes the Model-804 to leave the STANDBY state and reactivate the equipment. The sum-checks for system and user RAM are checked and if there is a fault a message will be displayed followed by a complete system test (see section 4.1).

Three basic operating modes are at the user's disposal.

KEY-mode (keyboard):	Operation of the Model-804 is via the built in keyboard, and output on the alphanumeric display (see chapter 5). KEY-mode is selected automatically after power-up.
REM-mode (Remote):	Operation of the Model-804 is from an external source (terminal, computer) through the RS-232C port (see chapter 7).
SIM-mode (Simulation):	In this mode the data stored in RAM is used to simulate an EPROM of an external unit. The connection is made via the simulator port (see chapter 9).

A LED is located on each of the MODE-keys. After pressing the appropriate MODE-key, the LED corresponding to the key pressed will illuminate, acknowledging your choice.

Any time a mode key is operated there is an overriding reset, even when the key pressed conforms to the system mode in use. Whenever you enter a mode, the system will reset to the start position for that particular mode.

3.2.3 HEXADECIMAL KEYBOARD (Group B)

The soft-touch hexadecimal keyboard is used to enter addresses and data. In KEY-mode the hexadecimal keyboard is also used to select a function, by entering the alphanumeric character adjacent to the function, on the hexadecimal keyboard.

3.2.4 CONTROL KEYS (Group C)

The control keys are used to respond to the various Model-804 prompts.

ENTR/INCR-key

The ENTR/INCR-key is used to confirm inputs, acknowledge outputs and terminate a procedure, such as inserting the EPROM. It is also used for incrementing the addresses for data input/output using the keyboard/display, continuing after a fault message and accepting default assignments.

NEXT/DECR-key

The NEXT/DECR-key is used for decrementing the addresses for data input/output via the keyboard display and for continuing operations after a fault message. This key is also used to scroll for the appropriate assignment during the initialization process.

CLR-key

The CLR-key can terminate various functions. The CLR key is used to abort a function while the equipment is awaiting an input. The CLR key may be disabled to avoid the danger of pressing it accidentally.

REPT-key

The REPT-key is used for repeating EPROM functions A thru D on the function list (group E Figure-1) with unchanged addresses and subfunctions. This is useful for repeating a given procedure on additional EPROMs.

3.3 OPERATING INDICATORS

Groups D and E of Figure-1 are indicators which will aid you in handling your Model-804. The operating condition indicators, and the list of functions are place-holders, designating the function that you are using, and the condition of the programmer.

3.3.1 OPERATING STATE INDICATORS (GROUP D)

These LEDs indicate the operating state of the Model-804. The LEDs corresponding to the state of operation light automatically as the Model-804 performs its functions. Operating states are:

INPUT: The Model-804 is awaiting an execution command from the keyboard.

BUSY: Operation is in progress. No input is possible.

ERROR: A fault message is on the display. The Model-804 is waiting for the ENTR-key to be pressed to acknowledge acceptance of the message.

In the KEY-mode one of these three LEDs will always be lit.

3.3.2 THE FUNCTION LEDS (Group E)

The functions are listed in two columns under the headings RAM and EPROM. LEDs are conveniently located adjacent to each function. When a function is chosen (by pressing the character on the hexadecimal keyboard) the LED will illuminate and remain lit throughout the function.

In the RAM column the first group of three functions define the origin of the data to be loaded into RAM. The three possible origins are:

1. The keyboard
2. The EPROM
3. The external instrument connected via the serial port.

The second group of four functions (functions 4 through 7) are the activities which can be performed on data in RAM:

4. FILL CONSTANT
5. MOVE
6. SEARCH
7. SPLIT/SHUFFLE

Once the data has been entered and the functions have been performed, data can be viewed either:

8. On the display
9. On a remote terminal via serial port

In addition to providing a wide variety of functions that can be performed in RAM, the Model-804 offers many options for manipulating data in the EPROM. With your Model-804, several types of tests can be performed on the data by interactive cimmunication between the RAM and the EW4PROM.

To perform these options you will select the functions in the second column under the heading EPROM:

A. CHECK
B. VERIFY
C. PROGRAM
D. ERASE

This concludes the discussion of the operating controls and indicators on the front panel. You will now see that communication between the user and the Model-804 is interactive. As you enter data and commands your Model-804 will respond to them.

3.4 RESPONSES

Every function that you perform will be acknowledged by the Model-804, with either the illumination of a LED or by an audible signal.

3.4.1 LED INDICATORS

There are 24 LEDs, in three groups, on the front panel of the Model-804. The LEDs indicate:

- operating mode (Figure 1 Group A)
- operating state (Figure 1 Group A)
- function (Figure 1 Group E): When a function is chosen the appropriate LED in the function table lights and remains lit until the entire operation is completed.

- voltage overlaod: The overload LED is above the EPROM socket (refer to Figure-1). If an overload occurs the supply voltage will be disconnected from the EPROM and the LED will illuminate. Correct the condition that caused the overload and reset either with one of the MODE-keys or with the CLR-key.

3.4.2 AUDIBLE SIGNAL

As operations are accepted by the Model-804 they will be verified by an audible signal. There are two distinguishable audible signals:

- the display
- the RS-232C

The output device used is determined by the MODE key pressed, or the function selected.

3.5 INPUT/OUTPUT

Data can be input in three ways. For this purpose there are three I/O outlets on the front panel:

- the display
- the RS-232C
- The SIM connector

The output device used is determined by the MODE key pressed, or the function selected.

3.5.1 DISPLAY

The 16-position alphanumeric display outputs various data:

- RAM address and data appear on the right side of the display.
- EPROM address and data appear on the left side of the display.
- text outputs (e.g. fault prompts).
- variable outputs (e.g. sumchecks).

3.5.2 RS-232C SERIAL PORT

The RS-232C has a 25-pin D-type connector for interfacing with an external instrument. See Chapters 7 and 8.

3.5.3 SIM CONNECTOR

The SIM connector is a 37-pin D-type interface for the special simulator cable, used to connect the Model-804 to the equipment which will simulate the EPROM (see chapter 9). This simulator cable is available for all standard EPROMs (see section 2.6).

3.5.2 RS-232C SERIAL PORT

The RS-232C has a 25-pin D-type connector for interfacing with an external instrument. See chapters 7 and 8.

3.5.3 SIM Connector

The SIM connector is a 37-pin D-type interface for the special simulator cable, used to connect the EP-804 to the equipment which will simulate the EPROM (see chapter 9). This simulator cable is available for all standard EPROMs (see section 2.6).

CHAPTER 4

GETTING STARTED

This section describes the operating steps after connecting the Model-804 to the main power. Figure 3 will help you follow the stages of the set up procedure.

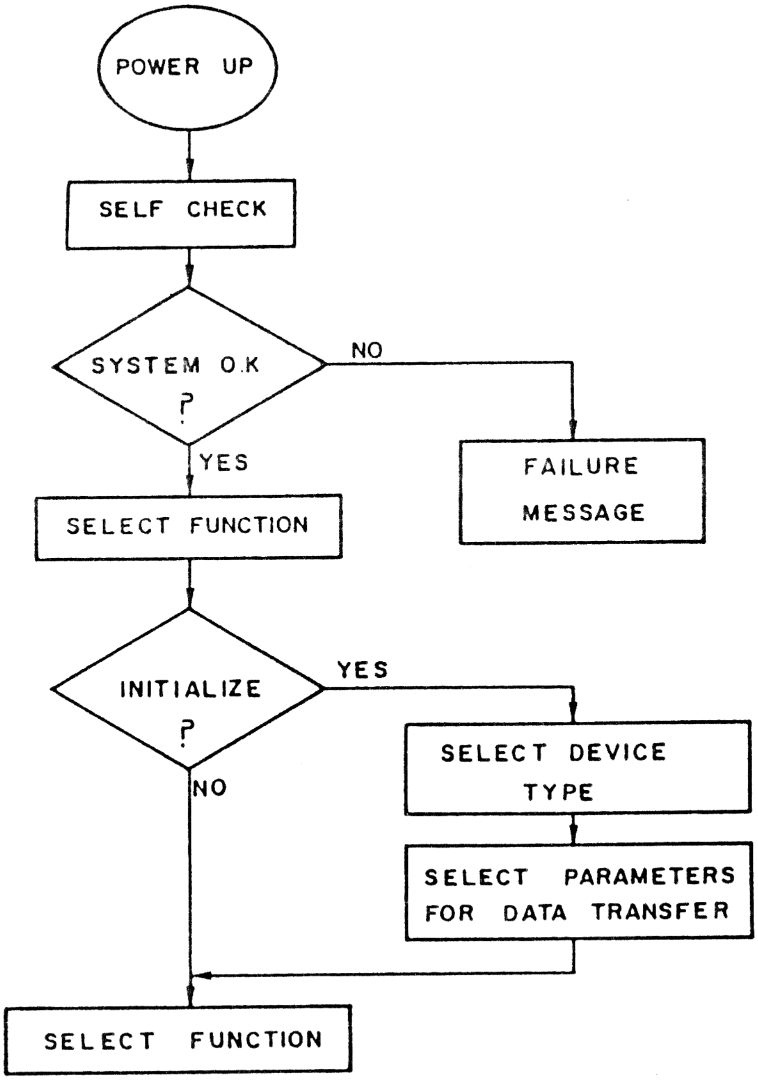


FIGURE-3 SET-UP PROCEDURE

4.1 SYSTEM TEST

When power is first supplied to the instrument the message

POWER FAILURE

will appear on the display, indicating that the user RAM is clear. Press ENTR to continue.

SELF CHECK

will then be displayed. This indicates that a system self-test is being performed.

If no fault is found the message

TEST OK mmmm VV

will be displayed letting you know that the equipment is in perfect working order. The display 'mmmm' equals the highest RAM address in your Model-804 and 'VV' is the software version. During the system check the data RAM is loaded with FF and the system RAM is loaded with the standard parameters.

If a fault is discovered any time that the programmer is in use the message

SYS ERR n mmmm

will be displayed. The letter 'n' is equal to the fault number and 'mmmm', any further information that may appear depending on the case. See Table-1 for a list of possible fault conditions.

TABLE 1

FAULT CONDITIONS

ERROR NUMBER	DESCRIPTION
0	Write error (verification after write to data RAM)
2	Sumcheck error in RAM data. The actual checksum is tested before execution of one of the functions 9, A, B, C and after power up.
4	Sumcheck error in RAM data (partial sumcheck). When an EPROM function is repeated due to REPT-key operation the data in the RAM between the predetermined address range is checked for a sumcheck error.
6	Sumcheck error concerning the system RAM after returning from STANDBY.
7	Sumcheck error in program memory. After power up (or after power failure) a sumcheck test is performed to check the system EPROMs.

4.2 FUNCTIONS WITHOUT INITIALIZATION

Most of the time you will be using the Model-804 with an EPROM, either loading data into or out of RAM. But part of the programming process may include manipulating data in RAM, independent of the EPROM. If you are planning to use the Model-804 to manipulate in RAM only, you are now ready to begin.

After the system test press ENTR to display

EPROM INT 2764

This is the default EPROM value, which can be ignored, as it is not relevant for functions bypassing the EPROM. Press ENTR to continue.

SELECT FUNCTION

will be displayed. You may now proceed by selecting the function desired.
See section 5.4 for a discussion of the functions performed in user RAM.

4.3 SELECTING THE DEVICE

Whenever you are performing functions involving an EPROM, the first step will be to let the Model-804 know which device you will be using. Selecting the device is the first step in function '0'.

After the TEST OK message press ENTR and the standard

EPROM INT 2764

will be displayed. Continue by pressing ENTR to display

SELECT FUNCTION

and select one of the function options. The programming algorithm for the EPROM INT 2764 is present in the software of your Model-804. Therefore if you are using this EPROM, you will not have to alter the parameters.

If, however, you are not using this EPROM, selecting the device will have to precede any further activity on the programmer.

The initialize function is labeled '0 INITIALIZE' on the function list (see Figure-1 Group E). When the message

SELECT FUNCTION

appears on the display, press '0' on the hexadecimal keyboard.

Note that after '0' is pressed the LED adjacent to the function '0 INITIALIZE' will illuminate and will remain lit throughout the function.

If the device is other than the EPROM INT 2764 it will be necessary to search for the appropriate device designation. The Model-804 offers three possible ways of selecting the device.

4.3.1 AUTOMATIC IDENTIFER

In the case of an inserted EPROM during type selection (Function 0) automatic type reading may be performed (Silcon Signature, Intelligent Identifer). Therefore, instead of a defined code number the value 'FF' (or 'xxFF') must be given and the type will be displayed in a mnemonic manner.

Since there are several manufacturers that do not implement this feature it can only be used if the currently selected type belongs to series 27xx and has 28 pins.

To initialize the Identifier enter the following:

1. Select FUNCTION 0.
2. The display will prompt: 'EPROM INT 2764'
3. Press NEXT, The display will prompt: 'EPROM BO'
4. Enter 'FF', press ENTER, the display will prompt (if EPROM has Automatice Identifier the type will be displayed). If there is no Automatic Identifer the display will prompt: 'NO AUTO IDENT'

4.3.2 ENTERING THE DEVICE CODE

It is possible to locate the device directly by entering the device code, as listed in Appendix A.

After SELECT FUNCTION press '0'. The first message will be

EPROM INT 2764

Press NEXT to change this value.

EPROM CODE XXXX

will be displayed instructing you to enter the code for the device that you will be using. You can either enter the correct code, as listed in appendix A, or you can press ENTR and scroll for the proper device.

4.3.3 SCROLLING TO LOCATE THE DEVICE

After you have pressed '0' on the hexadecimal keyboard, alternative devices will appear in sequence on the display. See Appendix-A for a listing of devices.

Use the NEXT-key to scroll to the appropriate device designation. When this has been located, press ENTR.

EXAMPLE:

Let us locate the EPROM MCM-68732. Listed under Motorola in Appendix A, the device code is 0050.

After system test the display will prompt the message

EPROM INT 2764

Press either ENTR or NEXT to display

SELECT FUNCTION.

Press '0' on the hex keyboard to enter the initialization mode.

EPROM INT 2764

will be displayed. Press NEXT to select a different EPROM type. The message

EPROM CODE XXXX

will be displayed. Enter '50' on the hex keyboard.

Scrolling to locate the device:

Alternatively you can scroll to locate this EPROM.

EPROM INT 2764

is the first option on the EPROM scroll, and the default EPROM value of the Model-804. When EPROM INT 2764 is displayed, instead of entering a device code, press NEXT to view the next EPROM choice.

EPROM INT 2764A

will be displayed. This is not the correct EPROM. Press NEXT to view the next EPROM and continue in this manner until

MCM68732

appears on the display. Press ENTR to acknowledge the correct display.

4.4 DEVICE ADAPTOR

The Model-804 can accomodate several EPROMs and EEPROMs for use with a device adaptor.

When a function has been selected for an EPROM requiring a DA, the Model-804 will prompt:

NEED ADAPT nnn

informing the operator which adaptor to use. The following is a list of adaptors and the devices that they support.

ADAPTOR	PROGRAMMABLE DEVICE
DA-8041	EPROM types 2815, and 2816.
DA-8042	EPROM type 8755A.
DA-8043	Single-chip microprocessor 8751, 8751H and 8716H.
DA-8044	EEPROM type 2817.
DA-8045	Single-chip microprocessor 874X
DA-8046	5V EEPROM types.

If you will be transmitting data either to or from the Model-804 via the RS-232C port, you will continue the initialization process and enter the parameters for transmission (section 4.5). If you will not be using the RS-232C port, you may exit the initialization function at this point by pressing CLR.

4.5 INITIALIZING PARAMETERS FOR TRANSMISSION

When performing functions involving the RS-232C port the parameters for transmission must be initialized.

The Model-804 will prompt with a list of parameters, and for every parameter a default value will be displayed. Alternative assignments are located by scrolling with the NEXT key. Correct assignments are acknowledged with the ENTR key, which will cause the next parameter and its default value to be displayed automatically.

Figure 4 will help clarify the system for selecting your EPROM, and its proper parameters. Table 2 will help clarify the process of selecting the parameters for transmission.

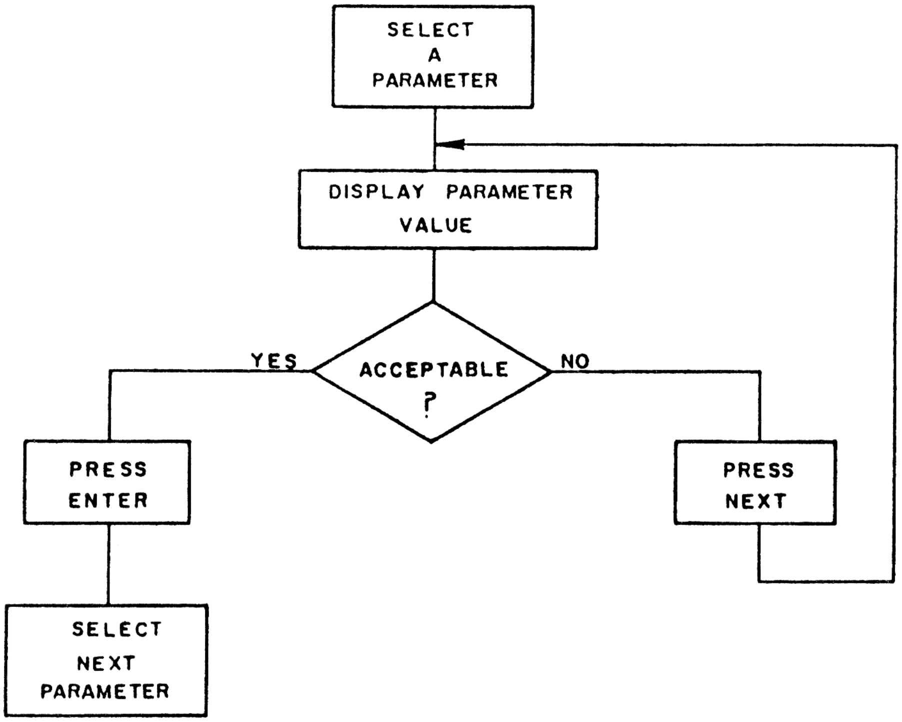


FIGURE-4 SELECTING A PARAMETER

TABLE 2

INITIALIZING THE PARAMETERS FOR TRANSMISSION

The Table below will help clarify the procedure for initializing the parameters for transmission. The first column at the far left is a listing of the parameters. The middle column shows exactly what will be displayed. The third column is an explanation of the parameter.

PARAMETER	FIRST DISPLAY	EXPLANATION
EPROM type	EPROM INT 2764	Selecting the device: All available EPROMs will be displayed sequentially in the form 'EPROM nnn' where 'nnn' is the EPROM type. See Appendix-A for a listing of all the programmable devices.
Address offset	ADDR-OFFSET 0	Selecting the RAM offset address.
Data format FORMAT	HEX-INTEL	Selecting the format: The available input/output formats for the RS-232C port will be displayed sequentially in the form 'format xxx' where 'xxx' is the format (see section 8.5).
Baud rate	BAUDRATE 9600	Selection of the baud rate: Baud rates will be displayed sequentially in the form BAUDRATE nnn' where 'nnn' is the baud rate in digits. Baud rate range is 50 to 19,200 (see section 8.3.2).
Stopbit	ONE STOPBIT	Selecting the stopbit: Alternate display of 'ONE STOPBIT' 'TWO STOPBIT' (see section 8.3.3).

Table 2 (continued)

Frame	7 DATA BITS	Selecting the Frame: Alternate display of '7 Data Bits' or '8 Data Bits' (see section 8.3.4).
Parity	ODD PARITY	Selection of parity: The text 'ODD PARITY', 'EVEN PARITY', 'SPACE PARITY' and 'NONE PARITY' will be displayed sequentially (see section 8.3.5).
Echo	AUTO ECHO	Selection of transmission mode: The text 'AUTO ECHO OFF' and 'AUTO ECHO ON' will be alternately displayed (see section 8.3.6).
Address source	ADDRESS FROM I/O	Selection of address source for communication formats. Alternate display of 'ADDRESS FROM I/O' and 'ADDRESS FROM KEY'(see section 8.3.7).
Address error	DISPLAY ADD ERR	Selection of preference for the display of address error: Alternate display of 'DISPLAY ADD-ERR' and 'IGNORE ADD-ERR' (see section 8.3.7).
Address display	RS232C DISPLAY ON	Selecting preference for viewing address changes during transmission. Viewing is appropriate for baud rates lower than 4800. For faster transmission rates, display must be OFF or transmission errors will occur (see section 8.3.8).

The initialization made at this point remains valid until there is a reinitialization or power is switched off. The Model-804 can be reinitialized by either:

- returning to the INITIALIZE function using the '0' key on the hex keyboard
- using the relevent command in REM-mode

Entering the STANDBY-mode or changing the operating mode does not change the initialization values. A power failure will cause the loss of the initialization setting, and parameters will reset automatically to the default values.

When the initializing process has been completed, the message

SELECT FUNCTION

will appear on the display. You will notice that the Model-804 goes to KEY-mode automatically after start-up. Note the illumination of the LED at the KEY-mode key.

If you are working either at a remote terminal or in simulation mode, now is the time to inform the Model-804 by pressing the appropriate key (REM or SIM). Your choice will be acknowledged by the illumination of the LED at the key of your choice. See chapters 7 and 8 for working in the remote mode, and chapter 9 for working in SIM.

CHAPTER 5

KEY MODE

5.1 THE FUNCTIONS

The functions are the tools for accomplishing the wide range of activities afforded by your Model-804. Programming is always executed in two steps. First, data is loaded into the user memory (RAM) and modified according to the user's needs. Secondly data is programmed from RAM to the EPROM. A host of functions are provided to execute this process. This chapter is a complete discussion of all of the functions in both RAM and the EPROM.

The functions that can be performed on the Model-804 are listed in two columns under the heading RAM and EPROM. Locate these functions in Figure-1, group E. For more detail see Figure-5 below.

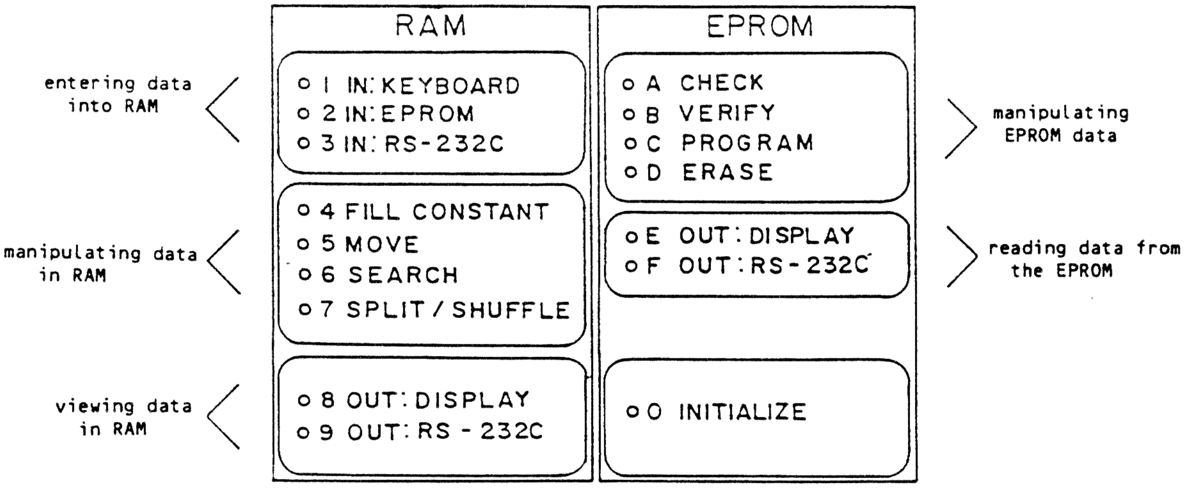


FIGURE-5 THE FUNCTIONS

In the column to the left under the heading RAM are three groups of functions for manipulation of data in RAM. The first three functions signify the ways of entering data into RAM. The second group are the functions for manipulating data in RAM, and the third group are the various ways that data in RAM can be viewed. In the second column under the heading EPROM, find the functions used for interaction with an EPROM, and below them the alternatives for reading data from the EPROM.

In the following section these functions will be discussed, with examples, in the order that they appear on the front panel.

5.2 SELECTING A FUNCTION

To select a function press the character on the hexadecimal keyboard corresponding to the desired function.

EXAMPLE:

If you will be using the function 'FILL CONSTANT' press '4' on the hex keyboard. If you will be using the 'MOVE' function press '5' on the hex keyboard, etc.

Note that when a function is chosen the LED adjacent to it on the list of functions illuminates, and remains lit throughout the process. The LED goes off when the function has been completed.

5.3 LOADING DATA INTO RAM

Referring back to figure-5, in the column under the heading RAM, you will see three groups of functions. In the first group, the functions

1 IN: KEYBOARD
2 IN: EPROM
3 IN: RS-232C

signify the source of the data to be loaded into RAM. Section 5.3 is a complete discussion of the proceedure for entering data using these three functions.

5.3.1 IN: KEYBOARD (function 1)

Function 1 is used to load the data into RAM manually.

After pressing '1' on the hex keyboard, the Model-804 will prompt

```
RAM BEGIN     0
```

The default address '0' can be changed by entering your start address on the hex keyboard. When the RAM begin address is displayed correctly press ENTR, and the message

DATA X YY

will be displayed. 'X' is the address entered. 'YY' equals the data contents of address 'X'.

Data addresses can be incremented and decremented using the ENTR and the NEXT-keys. If either the ENTR or NEXT-key is pressed longer than approximately one second the address will be continuously incremented/ decremented at the rate of about 8 addresses per second.

Enter data using the hex keyboard and store the displayed data byte with the ENTR key. After ENTR, the display will automatically prompt the next data address. When you are finished entering data, press CLR to return to SELECT FUNCTION.

EXAMPLE:

We are going to enter the following data at the corresponding addresses manually, using the the IN: KEYBOARD function:

Address	Data
01	1F
06	1E
07	1D
08	1C

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter 1	This number corresponds to IN: KEYBOARD on the function list.
2. RAM BEGIN 0	Enter 1	'0' is the default address which we will change to our begin address, '1'.
3. RAM BEGIN 1	Press ENTR	Press ENTR to confirm the correct display.

4. DATA 1 FF	Enter 1F	Thus we have changed our data at address '1' to '1F'.
5. DATA 1 1F	Press ENTR	Confirm the correct display.
6. DATA 2 FF	Press ENTR	The ENTR-key is used to increment addresses. Keep the ENTR-key pressed and increment to address '6'.
7. DATA 6 FF	Enter 1E and then ENTR	We have changed the data at address '6' to '1E' and confirmed with ENTR. Continue in this manner through address '08'.

5.3.2 IN: EPROM (function 2)

Function 2 is used to load the data from an EPROM into the user RAM.

The preliminary steps for working with this function are

- selecting the EPROM
- inserting the EPROM

Before continuing see section 4.3 on 'Selecting the EPROM', and Chapter 6 'Inserting the EPROM'.

After pressing '2' on the hexadecimal keyboard the EPROM that you have selected will be displayed. Continue by pressing either ENTR or NEXT.

RAM BEGIN 0

will automatically appear. The default address '0' can be changed by entering the correct figure on the hex keyboard. When the correct RAM begin address is displayed, press ENTR.The message

RAM END XXX

will be displayed. 'XXX' is the value for the highest address in the selected EPROM. This figure will vary depending on the device in use. Again, the correct address is acknowledged with the ENTR key.

Now that the destination addresses in RAM have been determined, it is necessary to designate the source addresses from the EPROM. The display will prompt

PROM BEGIN 0

Again the default value '0' is displayed, and can be changed via the hex keyboard. After ENTR the function will be executed.

After the data transfer, there will be an automatic verification of the EPROM and RAM data, which may generate an error message. A RAM checksum in the range between RAM BEGIN and RAM END will be calculated and displayed in the form 'CHKSUM nnnn'.

EXAMPLE:

In this example we will be copying a 100 byte block of data from the EPROM begining at address 500, into RAM address 200 to 2FF. We are using the EPROM 2732, which we have already initialized.

After selecting function 2 by entering '2' on the hex keyboard, the following prompts will be displayed.

DISPLAY	RESPONSE	EXPLANATION
1. EPROM 2732	Press ENTR	Acknowledge the correct EPROM with ENTR.
2. RAM BEGIN 0	Enter 200	Correct the default RAM begin address '0' by entering '200' on the hexadecimal keyboard.
3. RAM BEGIN 200	Press ENTR	We will acknowledge the correct RAM BEGIN address with ENTR.
4. RAM END FFF	Enter 2FF	The Model-804 will always prompt the largest address of the device used. The EPROM 2732 is a 4K-byte device. Hence its largest address is FFF. Since we are not interested in the

		entire EPROM capacity, only the 100 byte block, change the RAM END address to 2FF using the hex keyboard.
5. RAM END 2FF	Press ENTR	Acknowledge the correct display with ENTR.

Now the Model-804 will receive the addresses in the source block:

6. PROM BEGIN 0	Enter 500	Change the begin address in the EPROM using the hex keyboard.
7. PROM BEGIN 500	Press ENTR	It is not necessary to enter the end-address of the data block. The Model-804 will calculate this independently, based on the RAM addresses supplied above.

5.3.3 IN: RS-232C (function 3)

Function 3 is used for loading the RAM with data from a remote computer via the serial port. See chapter 8 for details on the Serial Port, including the different data formats, described in detail in section 8.5. Note the different address handling if the data format for development systems is selected.

After pressing '3' on the hex keyboard the Model-804 will prompt

RAM BEGIN 0.

The default RAM address '0' can be changed using the hex keyboard. When the correct address appears on the display press ENTR.

If you are using an Model-804 with 8K bytes,

RAM END 1FFF

will be displayed. This end address will vary, depending on the size of the memory in your Model-804. An Model-804 with 32K bytes will prompt with the maximum RAM address 7FFF.

The RAM END address determines the maximum size of the block of data that can be loaded into the Model-804 from a remote source. Loading a data block larger than 2000 in an 8K Byte system will cause an error message.

After entering the correct RAM END address press either ENTR or NEXT and the Model-804 will return to the begin address, where it will start receiving data from the external source.

This function can be ended in either one of the following ways:

- -reaching the RAM end address
- -pressing the CLR key
- -receiving the character ETX (Control C) from the serial port
- -receiving 'end block' in development system format
- -if a port error or address error occur (see chapter 6.3)

EXAMPLE:

In this example we will be transmitting a data block of 2K bytes via the RS-232C serial port, into the Model-804 (with 8K byte standard RAM). The destination address is 0 to 7FF.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter 3	Number '3' corresponds to the IN: RS-232C function.
2. RAM BEGIN 0	Press ENTR	Confirm the RAM begin address with ENTR.
3. RAM END 1FFF	Enter 7FF	'1FFF' is the highest address of the Model-804. Change the end address to 7FF using the hex keyboard.
4. RAM END 7FF	Press ENTR	The Model-804 will then return to the first address to wait for transmission.

5.4 FUNCTIONS PERFORMED IN RAM

Once the data block has been entered into your Model-804 (section 5.3) you can take advantage of your programmer's versatility in manip- ulating the data in RAM.

The next four functions, as listed on the front panel (see figure-5)

4 FILL CONSTANT
5 MOVE
6 SEARCH
7 SPLIT/SHUFFLE

are performed exclusively in RAM and are executed independent of an EPROM or an instrument connected at the serial port. Therefore there is no need to initialize before performing these functions.

These functions are especially useful as they enable the isolation of a block in RAM for manipulation, without affecting the rest of the RAM data. In the following section each of these functions will be discussed separately, and examples will be provided.

In brief, the function FILL CONSTANT (4) is used to fill a block with a data constant. The MOVE function (5) is used to move a data block to another address (without affecting the original block) and SEARCH (6) is used to locate a specific data byte. SPLIT and SHUFFLE are used to divide or to merge a data block into high and low addresses.

In the rest of this chapter the functions will be described in general terms, with an explanation of the prompts as they appear on the display. An example will follow the general discussion of each function. Follow the example for a step-by-step listing of the procedure for each function, from the first prompt, SELECT FUNCTION, through execution.

5.4.1 SPECIFYING THE DATA BLOCK

The first step in each of the functions in RAM is to inform the Model-804 which portion, by address, you will be handling. In this manual we will call that process "Specifying the Data Block."

Since this process must be completed in all of the functions, it will be described once below, and will not be repeated in the discussion of each function. Refer back to this section when needed. The prompts for specifying the data block will be described in the example for each function.

The message

RAM BEGIN 0

will be displayed. The default address value '0' can be changed via the hex keyboard. When the correct RAM BEGIN address is displayed press ENTR to acknowledge.

The last address in the RAM subset will then be suggested with a default address equal to the amount of RAM present in the device. If you are using an Model-804 with 8K Bytes

RAM END 1FFF

will be displayed. If you are using an Model-804 with 32K Bytes the RAM END address will read '7FFF'.

Enter the correct end address for the subset, and as before, acknowledge with the ENTR key.

EXAMPLE:

Let us specify 100 bytes of data in RAM, begining with address 50. We are working with the standard Model-804 with 8K Bytes.

DISPLAY	RESPONSE	EXPLANATION
1. RAM BEGIN 0	Enter 50	Change the default '0' using the hex keyboard. Since 50 is our start address, enter '50' on the the hex keyboard.
2. RAM BEGIN 50	Press ENTR	Confirm the address with ENTR.
3. RAM END 1FFF	14F	1FFF is the highest address in the RAM of our Model-804 with 8K byte. Since we are not interested in the entire memory capacity of the programmer, just in the 100 byte data block that we wish to isolate, we enter '14F' using the hex keyboard.

4. RAM END 14F Press ENTR Confirm with ENTR.

The data block for manipulation has now been defined.

In the examples for the next three functions

4 FILL CONSTANT
5 MOVE
6 SEARCH

we will follow through a single process of filling addresses 0 through 10 with the constant '89', then moving (function 5) that data block to addresses 50 to 60, and then searching (function 6) data block 0 to 60 for the blank addresses (data FF).

5.4.2 FILL CONSTANT (function 4)

Function 4 is used to load a RAM address range with a constant data word.

The first prompts will define the addresses in RAM to be filled. See section 5.4.1, 'Specifying the Data Block', for the details of this process.

When the subset has been defined the Model-804 will automatically prompt the message

DATA XX.

'XX' is the data value to be filled. This value can be changed using the hexadecimal keyboard. Press ENTR and the function will be executed. The RAM address range will be filled with the displayed value.

EXAMPLE:

In this example we will be filling the block address 10 to 20 with the constant '89'. We are using the basic Model-804 with 8K Bytes.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter 4	Number '4' corresponds to the function FILL CONSTANT.
Specifying the data block:		
2. RAM BEGIN 0	Enter 10	Our data block begins at address 10, hence we will enter '10' on the hex keyboard.
3. RAM BEGIN 10	Press ENTR	Acknowledge the correct address.
4. RAM END 1FFF	Enter 20	Remember that we are using an Model-804 with 8K Bytes. Thus the highest address is 1FFF. Since we will not be using the entire capacity for our function, we will change the end address to '20' using the hex keyboard.
5. RAM END 20	Press ENTR	Acknowledge with ENTR.
Entering data:		
6. DATA FF	Enter 89	We will be filling our address block with the constant '89'.
7. DATA 89	Press ENTR	The function will now be executed.

5.4.3 MOVE (function 5)

Function 5 is used to copy a data block in RAM. As in all functions the first step is to define the data block to be moved. See section 5.4.1 'Specifying the Data Block' for this procedure.

When the data block has been specified the Model-804 will prompt for the destination address.

DESTINATION 0

will be displayed. Enter the proper destination address, press ENTER to confirm, and the data of the source range will be loaded into the destination range.

Note: The data in the source address block will remain unchanged.

EXAMPLE:

We will be moving the address block '0' to '10' to fill addresses '50' to '60'. We are using the standard Model-804 with 8K Bytes.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter 5	We inform the Model-804 that we will be working in the MOVE function.
Specifying the data block:		
2. RAM BEGIN 0	Press ENTR	Since we will be moving an address block beginning with address '0', we will confirm the address displayed.
3. RAM END 1FFF	Enter 10	1FFF is the highest address in the RAM of our Model-804. Since our address block ends with address '10', enter '10' using the hex keyboard.
4. RAM END 10	Press ENTR	Acknowledge the correct display.
Entering destination:		
7. DESTINATION 0	Enter 50	The destination of our data is address 50.

8. DESTINATION 50	Press ENTR	It is not necessary to enter the last address of the destination. The Model-804 will calculate this independently, based on the RAM END addresses supplied at the beginning of the function.

With subset addresses and destination entered, the function will now be executed. Note that when the MOVE function is executed the original data block is not affected. The identical data now resides in blocks 0-10 and 50-60.

5.4.4 SEARCH (function 6)

Use function 6 to locate a data constant within a block of RAM.

The first procedure in this function as in all RAM functions, is defining the data block to be searched. See section 5.4.1 'Specifying the Data Block' for the first prompts of this function.

Having completed that procedure the Model-804 will request the data constant to be searched.

```
DATA   XX
```

will be displayed. Enter your data word using the hex keyboard. Press ENTR when the proper data is displayed.

The specified address range will then be compared to the data in RAM. At the first occurance of the constant within the data block, the message

```
FOUND   XXXX   YY
```

will be displayed. 'XXXX' is the data address, and 'YY' the data. If there is no match the message NOT FOUND will appear.

EXAMPLE:

For this example we will search the data block 0 to 60, for the data FF.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter 6	Enter the SEARCH function.
Specifying the data block:		
2. RAM BEGIN 0	Press ENTR	Confirm the begin address with ENTR.
3. RAM END 1FFF	Enter 60	1FFF is the highest address value in the Model-804 with 8K byte. We will enter '60' as our RAM end address.
4. RAM END 60	Press ENTR	Acknowledge the correct display.
Entering the data constant:		
5. DATA 00	Enter FF	We are searching for the blank addresses in the block. Hence we enter the data 'FF'.
6. DATA FF	Press ENTR	Acknowledge the correct display.
7. FOUND 11 FF	Press ENTR	The data 'FF' was found at address 11. We press ENTR, in response, to examine the next blank address.
8. FOUND 12 FF	Press ENTR	Continue in this manner. The Model-804 will return to SELECT FUNCTION when it reaches the highest RAM address.

5.4.5 SPLIT/SHUFFLE (function 7)

Use function 7 to perform either one of two activities: divide a data block into high and low addresses, or merge two blocks.

Display either 'SPLIT' or 'SHUFFLE' by pressing the NEXT-key, and acknowledge your choice of subfunction with ENTR.

SPLIT:
Splits odd and even address bytes, dividing them into two blocks stored in the upper and lower half of the memory, occupying the same amount of RAM. The center address will be the begin address of the second block and will be displayed following the RAM end address.

SHUFFLE:
Shuffles the block of RAM addresses by selecting data in an alternating pattern between the upper and lower data blocks, beginning with the first data address, creating one data block twice the size.

In either subfunction the first stage is to specify the data block to be manipulated. See section 5.4.1 'Specifying the data block' for the first prompts. The Model-804 will then execute the function. In both functions, when the process has been completed the display will prompt the center address

CENTER ADDR XXX.

If the input address range is not even, or if begin and end addresses are both odd or even, the message ADDR ERROR ODD will be displayed.

EXAMPLE:

Let us use the SPLIT function to divide the data block 34A through 353 into odd and even addresses.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Press 7	Enter the function SPLIT/SHUFFLE.
Specifying the data block:		
2. SPLIT	Press ENTR	Acknowledge the correct display.
3. RAM BEGIN 0	Enter 34A	This is our RAM BEGIN address.

4. RAM BEGIN 34A	Press ENTR	Confirm this display.
5. RAM END 1FFF	Enter 353	1FF is the highest address on our 8k 804. Enter our RAM END address.
6. RAM END 353	Press ENTR	Confirm the correct display.

Note: If the begin address is even, make sure that the end address is odd by adding an address. This is necessary because if the begin and end addresses are both even you will get an error message.

Executing the function:

7. CENTER ADDR 34F	The Model-804 will then divide the data block into two groups, one odd addresses the other even addresses, and will display the center address.

The data will be in the following configuration:

Addr.	Original	after SPLIT	after SHUFFLE
34A	AA	AA	AA
34B	BB	CC	FF
34C	CC	EE	BB
34D	DD	00	00
34E	EE	22	CC
34F	FF	BB	11
350	00	DD	DD
351	11	FF	22
352	22	11	EE
353	33	33	33

Note: The shuffle is performed on the original data, not on the data after the split.

This completes the discussion of the functions in RAM. Use the next two functions to view your data either on the display, or transmitted back to the remote computer.

5.5 VIEWING THE CONTENTS OF RAM

Referring back to figure-5, the last group in the column under the heading RAM are functions used to output data. The following two functions are used to view data either on the display or to the screen of your remote computer.

8 OUT: DISPLAY
9 OUT: RS-232C

5.5.1 OUT: DISPLAY (function 8)

Use function 8 to output the data from RAM onto the display.

The display will prompt

RAM BEGIN 0.

The address can be changed via the hex keyboard. Press ENTR to confirm the correct address.

Addresses can be incremented with the ENTR-key or decremented using the NEXT-key. If either key is held longer than approximately one second, the address will be continuously incremented/decremented at a rate of about eight addresses per second.

EXAMPLE:

Let's read the contents of addresses '34A' through '353' (assume that the data is the same as in the example for the SPLIT function in the previous section).

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter 8	Enter the OUT: DISPLAY function.
2. RAM BEGIN 0	Enter 34A	Enter our RAM BEGIN address.
3. RAM BEGIN 34A	Press ENTR	Confirm the correct display.

4.	34A AA	Press ENTR	Data 'AA' resides at address '34A'. Press ENTR to view the next address in sequence.
5.	34B CC	Press ENTR	Continue in this manner through the remainder of the address block.

5.5.2 OUT: RS-232C (function 9)

Use function 9 to output data from RAM via the serial port. To perform this function the data format and the parameters must be defined in the initialization process to correspond to the external device.(see section 7.6)

The display will prompt for the first and last addresses in RAM. Refer to section 5.3.1 'Specifying Data Block'.

When the correct begin and end addresses have been acknowledged with the ENTR-key the system will perform a self-check. The message

SELF CHECK

will be displayed.

Return to function input will occur when:

- -the end address is reached
- -the CLR-key is pressed
- -the character ETX (control C) is received.

EXAMPLE:

Let us output the data, at address 34A through address 353, to a remote computer via the serial port.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Press 9	'9' corresponds to OUT: RS-232C on the hex keyboard.

Specifying the data block:

2. RAM BEGIN 0	Enter 34A	Enter our RAM begin address.
3. RAM BEGIN 34A	Press ENTR	Confirm the display.
4. RAM END 1FFF	Enter 353	Enter our RAM END address.
5. RAM END 353	Press ENTR	Confirm the display.

Function execution:

6. SELF CHECK		The system will then perform a self-check.

This completes the discussion of the functions in RAM. In the examples for this section we filled various data blocks in RAM. The RAM of our Model-804 now has addresses 10 through 20 and addresses 50 through 60 filled with the constant '89'. Now we shall turn to the EPROM functions, and program some of our data from RAM into an EPROM.

5.6 EPROM FUNCTIONS

The functions listed in the second function column, under the heading EPROM (see figure-5):

A CHECK
B VERIFY
C PROGRAM
D ERASE

are used for interaction between the Model-804 user RAM and the EPROM.

The first two functions, CHECK and VERIFY, are used to test the condition of the EPROM itself, and to determine if it can be programmed with the contents of RAM. The PROGRAM function is used for programming the contents of RAM to the EPROM.

This section on the EPROM functions will be in the same format as section 5.4 "The Functions in RAM." Each function will be described in general terms, followed by an explanation of the prompts. An example will then follow listing all the prompts from the first, SELECT FUNCTION, through the termination of the function.

In the examples for the next two functions we will run some tests in preparation for programming the data constant '89' from user RAM, into the address block 10-20 in the EPROM. For all of these examples we will be using the EPROM 2732. Recall that the constant '89' has already been entered into the address block 10-20 in RAM.

Two steps must precede working with these functions:

- Selecting the EPROM (see chapter 4)
- Inserting the EPROM (see chapter 6).

When performing a device function, a device must be inserted in the EPROM socket. If the device is not inserted the programmer will detect this and the message

INSERT EPROM

will be displayed. Insert the EPROM and press ENTR to continue.

Note: When using a CMOS device the programmer will respond as though the device has not been inserted. Press ENTR to continue.

5.6.1 CHECK (function A)

The function CHECK is the only Model-804 function containing sub-functions. Under function A three different types of checks can be performed:

- BLANK CHECK
- SUM CHECK
- ILLEGAL BIT TEST

BLANK CHECK is used to check that either part or all of the EPROM addresses are blank. SUMCHECK is used to calculate a sumcheck on the given data block. These procedures are relevant to the EPROM exclusively, and are independent of RAM.

The third subfunction, the ILLEGAL BIT TEST is used to verify that data in RAM can be programmed on a preprogrammed EPROM. The ILLEGAL BIT TEST is an interactive process between the EPROM and the RAM.

Accessing the Subfunction

After pressing 'A' on the hex keyboard to enter the CHECK function, the EPROM that has been initialized will appear on the display. Confirm by pressing ENTR.

The subfunction

BLANK CHECK

will be displayed. This is the first of the three subfunctions. If you will be performing the blank check, press ENTR to acknowledge. Otherwise press the NEXT-key to scroll to SUMCHECK or ILLEGAL BIT TEST. Press the ENTR-key to accept the displayed subfunction.

Note: The subfunction performed last, will be displayed first, after pressing 'A'. A power-down will reset the scroll to the original position.

Specifying the Data Block

The first two subfunctions, BLANK CHECK and SUMCHECK, are tests performed exclusively on EPROM data blocks. The first stage in both subfunctions is to specify the data block (in the EPROM) to be checked.

Adjust the address for the prompt

PROM BEGIN 0

using the hex keyboard and confirm by pressing ENTR. The display will automatically prompt

PROM END XXX

which can be changed on the hex keyboard and confirmed with ENTR.

5.6.1.1 BLANK CHECK

In this subfunction a test is run to make sure that all the data in a specified address range is erased.

Refer to section 5.6.1 on accessing the subfunction and specifying the data block.

When the data block has been defined the function will be executed and either the message 'ERASED' or 'NOT ERASED" will be displayed. If the test concludes that the EPROM is not erased, the specific programmed addresses and data can be viewed by pressing the ENTR key.

EXAMPLE:

Let us check whether our EPROM is blank. Recall that we are using the EPROM 2732 (a 4K device) which is already inserted and initialized.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter A	'A' corresponds to the function CHECK.
2. EPROM 2732	Press ENTR	Confirm with ENTR.
3. BLANK CHECK	Press ENTR	Confirm with ENTR.
4. PROM BEGIN 0	Press ENTR	We are checking the entire address range of the EPROM, beginning with address 0. Therefore we will confirm the displayed address.
5. PROM END FFF	Press ENTR	The EPROM 2732 has 4k bytes. Therefore its highest address is FFF. As we are checking all address through FFF, press ENTR.

6. NOT ERASED	Press ENTR	Let us view which addresses in our data block are not erased.
7. 10 AB	Press ENTR	The first address not erased, '10', has the data content 'AB'.
8. 11 AB	Press ENTR	Address 11 has also been programmed with 'AB'. Continue in this manner until all the addresses that are not erased have been displayed.

This test will reveal that our EPROM is programmed with the constant 'AB' in the address block 10 to 20. The rest of the EPROM is in virgin state.

5.6.1.2 SUM CHECK

In this subfunction a 16-bit checksum for the specified EPROM address range is calculated.

See section 5.6.1 on accessing the subfunction and specifying the data block.

When the data block has been defined the function will be executed and the checksum for the given data block will be displayed in the format

CHKSUM XXXX.

EXAMPLE:

Let us perform a sumcheck of the EPROM address block 10-20. Recall that we are using the EPROM 2732 which is already inserted and initialized.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter A	Enter the CHECK function.
2. EPROM 2732	Press ENTR	Confirm with ENTER.

3. BLANK CHECK	Press NEXT	Scroll to the following subfunction with the NEXT key.
4. SUM CHECK	Press ENTR	Confirm with ENTR.
5. PROM BEGIN 0	Enter 10	The first address of our data block is 10.
6. PROM BEGIN 10	Press ENTR	Confirm with ENTR.
7. PROM END FFF	Enter 20	The last address in our data block is 20.
8. PROM END 20	Press ENTR	Confirm with ENTR.
9. CHKSUM 05D8		This is the checksum for our data block.

5.6.1.3 ILLEGAL BIT TEST

In this subfunction a comparison is made between every 0 bit in the data block in RAM and the EPROM to determine if the block in EPROM can be programmed with the data in RAM. Programming can be executed if every 0 bit in EPROM is also 0 in RAM.

To access the subfunction see section 5.6.1.

After pressing ENTR to acknowledge the subfunction the display will prompt for the first and last addresses in the data block in RAM.

RAM BEGIN 0

will be displayed. Change the begin address using the hex keyboard, and press ENTR to acknowledge the correct address.

RAM END FFF

will then appear. This address can also be changed using the hex keyboard.

Press ENTR to acknowledge the correct address. The Model-804 will then request the first address of the data block in the EPROM by displaying the message

PROM BEGIN 0

which can be changed using the hex keyboard, and confirmed with the ENTR key. The function will then execute. The message

SELF CHECK

will be displayed while the function is being performed. Following the execution the results of the test will be displayed. Either the message 'NO ILLEGAL BITS' or 'ILLEGAL BIT ERR' will appear.

EXAMPLE:

We will perform an illegal bit test on the EPROM data block 10-20 to check whether it can be reprogrammed with the data in RAM.

Recall that we are using the EPROM 2732 which is already inserted and initialized.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter A	'A' corresponds to the function CHECK.
2. EPROM 2732	Press ENTR	Confirm with ENTR.
3. SUM CHECK	Press NEXT	Scroll to the following subfunction by pressing the NEXT key.
4. ILLEGAL BIT TEST	Press ENTR	Confirm with ENTR.
5. RAM BEGIN 0	Enter 10	The first address of our RAM data block is 10, hence we will enter '10' on the hex keyboard.

6. RAM BEGIN 10	Press ENTR	Confirm with ENTR.
7. RAM END 100F	Enter 20	The last address in our RAM data block is 20, hence we will enter '20' on the hex keyboard.
8. RAM END 20	Press ENTR	Confirm with ENTR.
9. PROM BEGIN 0	Enter 10	The first address of our PROM data block is 10, hence we will enter '10' on the hex keyboard.
10. PROM BEGIN 10	Press ENTR	The Model-804 will calculate the end address of the PROM data block independently, based on the addreses of the RAM block.
11. SELF CHECK		The system compares the two blocks.
12. NO ILLEGAL BITS		Programming can be executed.

5.6.2 VERIFY (function B)

Use function B to verify that the data in EPROM and RAM are identical.

After pressing 'B' on the hex keyboard, the EPROM that has been initialized will appear on the display. Confirm the correct display with ENTR.

The display will then prompt for the first and last addresses in the data block in RAM.

RAM BEGIN 0

will be displayed. Change the begin address using the hex keyboard, and press ENTR to acknowledge the corrected address.

RAM END XXX

will then appear. This address can also be changed using the hex keyboard. Press ENTR and continue.

The Model-804 will then request the first address of the data block in the EPROM by displaying the message

PROM BEGIN 0

which can be changed using the hex keyboard, and confirmed with the ENTR key. The function will then execute. The message

SELF CHECK

will be displayed. If no discrepancies are found the message

VERIFY OKAY

is displayed. If the verification process reveals a fault, the display will prompt

COMPARE ERROR.

Press ENTR to view the addresses and words that do not correspond. The data word for the EPROM will be displayed on the left and the RAM content on the right side.

EXAMPLE:

Let us verify the address block 10-20 in RAM and the EPROM.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter B	'B' corresponds to the function VERIFY.

2. EPROM 2732	Press ENTR	Confirm with ENTR.
3. RAM BEGIN 0	Enter 10	The first address of our RAM data block is 10, hence we will enter '10' on the hex keyboard.
4. RAM BEGIN 10	Press ENTR	Confirm with ENTR.
5. RAM END FFF	Enter 20	The last address in our RAM data block is 20, hence we will enter '20' on the hex keyboard.
6. RAM END 20	Press ENTR	Confirm with ENTR.
7. PROM BEGIN 0	Enter 10	The first address of our PROM data block is 10, hence we will enter '10' on the hex keyboard.
8. PROM BEGIN 10	Press ENTR	The Model-804 will calculate the end address in the PROM data block independently, based on the addresses of the RAM block.
9. SELF CHECK		The system compares the two blocks.
10. COMPARE ERROR	Press ENTR	View the inconsistent addresses by pressing ENTR.
11. 10 AB 10 89	Press ENTR	The data 'AB' resides at address '10' in the EPROM, whereas in RAM address '10' contains the data word '89'. To view the next inconsistent address press ENTR, and continue throughout the address block.

5.6.3 PROGRAM (function C)

Use this function to program an EPROM with the data from RAM. The following is the sequence of operation that will take place internally when you are working in the PROGRAM function.

General operating sequence:

1. The illegal bit test. See section 5.6.1.3.
2. The programming is done according to the specifications supplied by the EPROM manufacturers.
3. The EPROM and RAM data are compared. See Verify, section 5.6.2.

1. The illegal bit test:

In the specified address range a check is made to see if any bits in the EPROM are already programmed (set to 1) preventing reprogramming with RAM data.

If a fault is found

```
ILLEGAL BIT ERR
```

is displayed.

2. Programming

The method of programming depends on the specifications supplied by the EPROM manufacturers and therefore differs for every type.

There are two main groups of programming algorithms for EPROMs

a. The whole address range is gone through many times (e.g. 50 loops).

   ```
   XXX  LOOP NUMBER
   ```

   will be displayed with the actual numbered loop being executed.No error checking will be done during programming.

b. The address range is gone through only once. The actual address and data from RAM and EPROM are displayed.

If there is a programming error the message

PROGRAMMING ERROR

will be displayed. Programming can be aborted with the CLR key without damage to the EPROM.

3. Verify:

The data in EPROM and RAM are compared and if no discrepancies are found the message PROGRAMMING OKAY is displayed. Otherwise COMPARE ERROR is displayed with the addresses in the EPROM and RAM which do not correspond. To continue viewing the inconsistent data use the ENTR-key. Abort with the CLR-key.

To program an EPROM with the data from RAM press 'C' on the hex keyboard to enter the PROGRAM function. The EPROM that has been initialized will appear on the display. Confirm with ENTR.

The display will then prompt the first and last addresses in the data block in RAM.

RAM BEGIN 0

will be displayed. Change the begin address using the hex keyboard, and press ENTR.

RAM END FFF

will then appear. This address can also be changed using the hex keyboard. Acknowledge with ENTR.

The Model-804 will then prompt the first address of the data block in the EPROM with the message

PROM BEGIN 0.

Change the address using the hex keyboard, and confirm with ENTR. The message

SELF CHECK

will be displayed. While the function is executed, the message

XXX LOOP NUMBER

or the current programmed address and data are displayed. The function is completed with the display

PROGRAMMING OKAY.

EXAMPLE

We are now ready to program our data constant '89' from the adddress block 10-20 in RAM, into address block 10-20 in the EPROM 2732. As in our previous examples, the EPROM is already inserted and initialized.

You will recall that the address block 10-20 is already programmed with data AB. The illegal bit test has confirmed that reprogramming with the data from RAM ('89') is legal.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter C	'C' corresponds to the function PROGRAM.
2. EPROM 2732	Press ENTR	Confirm with ENTR.
3. RAM BEGIN 0	Enter 10	Change the RAM BEGIN address to '10' using the hex keyboard.
4. RAM BEGIN 10	Press ENTR	Confirm with ENTR.
5. RAM END FFF	Enter 20	Change the RAM END address to '20' using the hex keyboard.

6. RAM END 20	Press ENTR	Confirm with ENTR.
7. PROM BEGIN 0	Enter 10	Change the begin address in the EPROM to '10'. The Model-804 will calculate the last address independently based on the RAM end address.
8. SELF CHECK		The function is being executed.
9. XXX LOOP NUMBER		
10. PROGRAMMING OKAY		The function is complete. The EPROM has now been programmed with the data from RAM.

5.6.4 ERASE

This function is used to erase EEPROMS. The Model-804 will check the initialization, and if there is an error the message

NOT ERASABLE

will be displayed.

Initialize and insert the EPROM. The message

SELECT FUNCTION

will be displayed. Enter the 'D' on the hex keyboard, and press ENTR to conmfirm. The function will be executed, and the message

ERASED

will be displayed.

EXAMPLE

Let us erase the EEPROM HN 48016. Insert and initialize the EEPROM.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Press D	'D' corresponds to the function ERASE.
2. EPROM HN 48016	Press ENTR	Confirm with ENTR.
3. ERASED		The function is complete, and the EEPROM is erased.

5.7 VIEWING THE DATA FROM THE EPROM

5.7.1 OUT: DISPLAY (function E)

Use this function to output the data from the EPROM to the DISPLAY.

The display will prompt the first address to be viewed by displaying the message

PROM BEGIN 0

The address can be changed using the hex keyboard. Press ENTR to confirm the correct address.

Addresses can be incremented and decremented using the NEXT-key. If either key is held longer than approximately one second addresses will be continuously incremented/decremented at a rate of about eight addresses per second.

EXAMPLE:

Let us view the contents of addresses 10-20 of our EPROM 2732, which has been inserted and initialized.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter E	'E' corresponds to the function OUT: DISPLAY.
2. EPROM 2732	Press ENTR	Confirm with ENTR.
3. PROM BEGIN 0	Enter 10	The first address to be viewed is '10'.
4. PROM BEGIN 10	Press ENTR	Confirm the correct display with ENTR.
5. 10 89	Press ENTR	The data word '89' resides at address '10'. Press ENTR to view the next address.
6. 11 89	Press ENTR	The data word '89' resides at address '11'. Press ENTR to view the next address and continue in this manner throughout the data block.

5.7.2 OUT: RS 232 C (function F)

Use this function to output the data from the EPROM via the serial port.
To perform this function the data format and the parameters must be defined in the initialization process (see section 4) to correspond to the external device (see section 7.6)

The first stage in this function is to specify the EPROM data block to be transmitted. Enter the correct address for the prompt

PROM BEGIN 0

and acknowledge with ENTR. The display will automatically prompt

PROM END XXX.

'XXX' is the highest address of the EPROM used. Enter the correct address and confirm with ENTR.

EXAMPLE:

Let us output the data block 10-20 from our EPROM 2732 via the serial port.

DISPLAY	RESPONSE	EXPLANATION
1. SELECT FUNCTION	Enter F	'F' corresponds to the function OUT: RS-232C.
2. EPROM 2732	Press ENTR	Confirm with ENTR.
3. PROM BEGIN 0	Enter 10	'10' is our begin address.
4. PROM BEGIN 10	Press ENTR	Confirm with ENTR.
5. PROM END FFF	Enter 20	'20' is our end address.

The function will then be executed.

CHAPTER 6

HANDLING THE EPROM

6.1 SELECTING AN EPROM

All of the EPROMS listed in Appendix A (a table of programmable devices) can be used with your Model-804.

When you have selected your EPROM, follow the directions in Chapter 4 for initializing.

Always be sure that the EPROM displayed at the start of the function, is the same as the one inserted. If there is a discrepancy remove the EPROM before reinitializing.

CAUTION: NEVER INSERT THE EPROM BEFORE INITIALIZATION.

6.2 INSERTING THE EPROM

Insert the EPROM in the socket labeled EPROM, located at the center of the Model-804. Refer to figure-1 to locate the socket on the front panel. Hold the EPROM with the notched side upwards, and insert the pins from the bottom up. A 24-pin EPROM will be placed so that the top four pins remain free.

The lever at the top left-hand side of the EPROM socket must be upright when placing the EPROM in the socket. Once inserted, lower the lever to lock the EPROM in place.

6.3 PRECAUTIONS

Observe the following precautions to avoid damaging the EPROM.

Remove your EPROM before:

- Inserting or removing the plug to the mains.
- Switching the power on or off with the MODE keys.
- Initializing a new EPROM type.

The supply voltages to the EPROM will be disconnected in the event of an overload. The OVERLOAD LED will illuminate and the function will be terminated. Use the CLR-key or a MODE-key to reset after a power overload.

digelec

PROM & LOGIC
PROGRAMMERS

PRODUCT RELEASE

NEW SOFTWARE REV 4.12 FOR MODELS 804/5

1. SCOPE

Software REV 4.12 is now available for models 804/5 E/EEPROM programmers. It offers first-time support for a wide range of devices and algorithm modifications for currently supported devices, an enhanced REMOTE mode, and various operational improvements.

2. NEW/MODIFIED DEVICE SUPPORT

Mfr.	Device Number	Device Code	DA	Remarks
AMD	AM27C256	00A9		Modified algorithm
	AM27C512	00AD		Modified algorithm
	AM27C1024	2048	DA-8048	Modified algorithm
Catalyst	CAT27HC256	0099		
Intel	D68C257	00DB		
	D27C010	2000		
National	NMC27C32B	0049		
	NMC27C256	00D8		
	NMC27C512	00DC		
	NMC27C020	2100	DA-8047	
	NMC27C2048	2140	DA-8048	
NEC	uPD27C2001	2104	DA-8047	
	uPD27C4001	2204	DA-8047	
RCA	CPD18U42C	1842	DA-3041	Custom made
Signetics	87C51	1553	DA-8043	Modified algorithm
Texas	TMS27C010	2026	DA-8047	
Toshiba	TC54512AP	009D		
	TC574000	2204	DA-8047	
WSI	WS27C040	2200	DA-8047	

3. REMOTE MODE

New commands and error bits designations increase REMOTE operating possibilities, as follows:

Command	Function
A	Defines data transfer format with the following format codes: 00 Manual 01 HEX Space 03 BNPF 10 HEX Intel 30 Extended HEX Intel 11 Motorola (S1) 31 Extended Motorola (S2) 12 Tektronix 13 MOS Technology 80 ceLink binary
]	Initiates automatic identification of inserted EPROM
#	Resets hardware for serial interface and overload circuit
p	Programs security bit of 8751 family (single-chip microprocessors)
&	Provides system error messages in format uWWWW/sXXXX/rYYYY/xZZ where: u = Unit (2000 = 804, 2100 = 805) s = Software revision number r = RAM size x = System error code

Error Code	Error Bit	Meaning
38	27	Command A sent without format code
99	28	Timeout error during data transmisssion in ceLink format
35	29	Command] sent when no EPROM inserted
40	30	<CLR> pressed on Model 804/5

4. MISCELLANEOUS

a. If you attempt to perform a PROGRAM, VERIFY or ILLEGAL BIT CHECK function on part of a device whose entire device size is larger than your programmer's RAM, the error message **LIMIT BY RAM SIZE** will no longer be displayed.

b. Before you turn off your programmer, the message **REMOVE EPROM** will advise you to remove any inserted device to avoid damaging it.

c. If during REMOTE mode idle state an overcurrent occurs at Model 804/5, a single F is sent to the programmer.

d. All device types are correctly displayed and entered.

e. When selected REMOTE mode, default initialization of RS232C interface is now EOF ON.

digelec

PROM & LOGIC
PROGRAMMERS

ERRATA

When connecting programmer models 803, 824, 825, 828, 834, and 860 to an IBM AT, for operation with or without the appropriate software program (SOFTLINK and LOGILINK), refer to the table below for the proper pin connections, and update your User Manual accordingly. Take particular note of CTS and RTS connections.

We are sorry for any inconvenience that this error may have caused.

IBM PC AT *Serial Port				Programmer Serial Port	Signal	Signal Name	Input/ Output
Rx	2	----\/----	2	Rx	Rx	Receive Data	Input
Tx	3	----/\----	3	Tx	Tx	Transmit Data	Output
RTS	8	----\/----	4	RTS	RTS	Request To Send	Output
CTS	7	----/\----	5	CTS	CTS	Clear To Send	Input
DTR	4	----------	6	DSR	DSR	Data Set Ready	Input
DSR	6	----------	20	DTR	DTR	Data Terminal Ready	Output
GND	5	----------	7	GND	GND	Signal Ground	--

*9-pin D-type connector

digelec

PROM & LOGIC

PROGRAMMERS

Rec'd
5/91
Proms now 4.15 ✓

Dear User:

Thank you for buying DIGELEC.

As a result of software upgrades, Chapter 7 in your User Manual has become outdated. Please replace it with the enclosed revised Chapter 7.

Alex Abelson

Alex Abelson
Quality Assurance

CHAPTER 7

REMOTE MODE

7.1 INTRODUCTION

REMOTE mode is designed to enable you to control EP-804/5 from a remote computer, connected by means of the RS-232C port.

Communication between EP-804/5 and the computer is interactive, initiated when you send commands from your remote computer instructing EP-804/5 to perform various functions. EP-804/5, in turn, transmits back characters in response to these commands, advising you of the outcome of a function, or of general operating status.

This chapter serves as a guideline to enable you to incorporate EP-804/5 remote commands into your computer operating software.

7.2 SETUP

You may use REMOTE mode with any computer that has an RS-232C port. To activate it, proceed as follows at your programmer station:

a. Press <REM> and <ENTR>. The following message will be displayed:

   STANDARD REMOTE

b. To accept this, press <ENTR>.

c. Other types available include Intel MDS, Philips PMDS I and II. To select one of these, press <NEXT> to display it, and <ENTR> to select it. The interface parameters will automatically be adjusted.

Set the following transmission parameters at your computer station:

Type	Display	Baud Rate	Parity
Standard	STANDARD REMOTE	*	*
REMTYP	REMTYP CEL SLOW	9600	*
REMTYP	REMTYP CEL FAST	19200	*
Intel MDS	INTEL MDS REMOTE	9600	Odd
Philips PMDS I	PMDS 1 REMOTE	9600	Even
Philips PMDS II	PMDS 2 REMOTE	4800	Even

* Set to match EP-804/5

7.3 COMMUNICATION PROTOCOL

7.3.1 AT COMPUTER STATION

Most REMOTE mode commands consist of a single ASCII character, followed by a <CR>. The following keys at your remote computer also affect Model 804/5 operation:

<ESC>	Aborts a command
<BREAK>	Aborts data transfer
<CR>	Signals completion of an action, both at programmer and computer stations

In cases where a command requires a parameter, it must conform with the following syntax:

XX...XXB, <CR>

where:
X = Any number of HEX characters, the last four of which are parameters (if fewer than four Xs, leading zeros are automatically added; if omitted, parameter is set to zero)
B = Command

7.3.2 AT MODEL 804/5

Model 804/5 responds to commands from the remote computer by transmitting back one of the following characters:

>	Indicates successful activation of REMOTE mode, when followed by <CR>, or successful completion of function
F	Precedes fault message. An error status word is sent as eight HEX characters (four bytes), where each bit represents an error code.
?	Indicates unknown or invalid command

7.4 SELECTING A DEVICE TYPE

The first step in beginning operation is selecting a device type. This must be done before you insert a device in the socket.

To select the device type in KEY mode at your programmer, use initialization function 0. You may also select the device type in REMOTE mode at your computer, by sending command @.

If you are in REMOTE mode and attempt to execute a function before selecting a device type and inserting the correct device, the following message will be displayed at your programmer:

INSERT EPROM

Proceed as instructed, and press <ENTR> to acknowledge and continue.

7.5 THE COMMANDS

The commands described below are grouped as discussed in Chapters 4 and 5, and are listed alphabetically within each group.

7.5.1 SETUP FOR TRANSMISSION

Command	Function	Description
cccc@	SELECT EPROM TYPE	**WARNING!** Before using this command, make sure there is no device in the device socket! Sets device type, where cccc = EPROM device code, which you may enter without leading zeros;
xxA	SET DATA TRANSFER FORMAT	Sets one of the following data transfer format codes (xx): 00 Manual 01 HEX Space 03 BNPF 10 HEX Intel 30 Ext. HEX Intel 11 Motorola (S1) 31 Ext. Motorola (S2) 12 Tektronix 13 MOS Technology 80 ceLink Binary

Command	Function	Description
D	SET ODD PARITY	**NOTE:** When altering the parity setting at your programmer, make sure that the setting at your computer coincides. Otherwise, communication may fail. Selects odd parity at programmer
E	SET EVEN PARITY	Selects even parity at programmer
G	SOFTWARE VERSION	Programmer transmits three-digit label indicating current software version
K	SET TWO STOP BITS	Sets two stop bits. Programmer responds by transmitting back two stop bits.
N	NO PARITY BIT	Selects no parity at programmer
aaaaW	SET ADDRESS OFFSET	Sets address aaaa, to be subtracted from all addresses transferred during data input through the serial interface, and to be added to addresses sent during data output
#	RESET	Resets hardware for serial interface and overload circuit

7.5.2 DATA TRANSFER

Command	Function	Description
I	INPUT PORT DATA TO RAM	Loads data into RAM from RS-232C port
L	INPUT EPROM DATA TO RAM	Loads data from programmed EPROM into RAM
O (Capital O)	OUTPUT RAM DATA TO PORT	Sends RAM data to computer
o (Lower case)	OUTPUT EPROM DATA TO PORT	Sends EPROM data to computer

7.5.3 RAM FUNCTIONS

Command	Function	Description
iiii<	SET RAM BEGIN ADDRESS	Sets RAM begin address to value iiii; default is zero
ssss;	DEFINE BLOCK SIZE	Defines block length ssss
_	CLEAR RAM	Fills RAM with a constant value, corresponding to erased state of device type, defined by address range character < (set RAM address) and ; (set block size)
C	RAM DATA COMPARED TO PORT DATA	Determines if discrepancy between data from port and RAM data (if yes, transmits F back to computer)
J	SPLIT	Performs split operation on device data on address range defined by characters < (set RAM address) and ; (set block size)
j	SHUFFLE	Performs shuffle operation on device data on address range defined by characters < (set RAM address) and ; (set block size)
S	RAM CHECKSUM	Calculates RAM checksum and transmits 16-bit result back to computer in HEX form xxxx. Useful for checking correct loading of data from computer.

7.5.4 EPROM FUNCTIONS

Command	Function	Description
bbbb:	SET EPROM BEGIN ADDRESS	Sets EPROM begin address to value bbbb; default is zero.
[	OUTPUT OF EPROM DEVICE TYPE	Requests EPROM device type, which programmer transmits back in form nnnn
]	EPROM AUTO MATCH	Instructs programmer to automatically identify device type of inserted EPROM
B	BLANK CHECK	Checks EPROM to determine if in erased state

Command	Function	Description
P (Capital P)	PROGRAM EPROM	Performs three programming stages: illegal bit test, programming, verifying; programmer transmits back F (fail) character if any stage fails, and > if entire process successful
p (Lower case)	PROGRAM 8751 SECURITY BIT	Programs security bit of 8751 family of single-chip microprocessors
R	REQUEST EPROM STATUS WORD	Programmer transmits back EPROM status word in form aaa/b/c or aaaa/b/c, where: aaa/a = Maximum EPROM address in HEX; (f < 1000H, three places are used; otherwise, four) b = Number of EPROM data bits (8) c = Erased state of EPROMs (0)
s (Lower case)	EPROM CHECKSUM	Calculates EPROM checksum and transmits 16-bit result back to computer in HEX form xxxx.
T	ILLEGAL BIT CHECK	Checks if programmed device can be programmed with RAM data
V	VERIFY DATA	Compares EPROM and RAM data

7.5.5 ERRORS

Command	Function	Description
&	SYSTEM STATUS MESSAGE u2000 ✓ s1100 ✓ r007F	Programmer transmits system status message in form uWWWW/sXXXX/rYYYY/xZZ where: u = Unit (2000 = 804, 2100 = 805) s = Software revision number r = RAM size x = System error code
F	REQUEST ERROR STATUS WORD	Programmer transmits error status word in form xxxxxxxxx, where x = HEX character, and resets all bits to 0
X	REQUEST ERROR CODES	Programmer transmits error code in form XX, where X = decimal character strung together without separation
Y	REQUEST NUMBER OF PARITY ERRORS	Programmer increments internal counter if parity error found, and transmits this number to computer

7.6 ERROR HANDLING

7.6.1 ERROR STATUS WORD

Errors occurring during operation in REMOTE mode are recorded in the error status word, which can be retrieved by using the command F. The bits have the following designations:

Bit	Designation
31	Any error
30	Always 0
29	Always 0
28	Always 0
27	Always 0
26	Overrun error (character cannot be processed)
25	Framing error (no stop bit)
24	Always 0
23	Any EPROM error
22	Overload
21	No EEPROM type selected (command [)
20	EPROM device type number invalid (command @)
19	EPROM not erased
18	EPROM not programmable (only with function T)
17	EPROM data does not correspond with RAM data
16	EPROM not programmable (only with function P)
15	Any serial port error
14	Always 0
13	Always 0
12	Port data does not correspond with RAM data
11	Port checksum error
10	Port data word error
9	Port address error
8	Port character error
7	Any RAM error
6	Always 0
5	Always 0
4	Address range too small (only 3 V EPROMs)
3	RAM/EPROM address error
2	Always 0
1	RAM write error
0	Always 0

7.6.2 ERROR CODES

When it receives the command X, MOdel 804-5 generates error codes based on the bits of the error status word. The codes have the following meanings:

Code	Bit	Meaning
20	19	EPROM not erased
21	18	EPROM not programmable (only with function T)
23	17	EPROM data does not correspond with RAM data
23	16	EPROM not programmable (only with function P)
27	3	RAM, EPROM address error
28	2	Odd address for commands J and j
29	12	Port data does not correspond with RAM data
31	22	Overload
34	21	No EPROM type selected (only with function x)
34	20	Invalaid EPROM type number (only lwith function @)
35	29	Command j sent when no EPROM inserted
36	4	Address range too small (only 3 V EPROMs)
38	27	Command A sent without format code
40	30	<CLR> pressed
48	26	Overrun error (character cannot be processed)
48	25	Framing error (no stop bit)
63	1	RAM write error
81		Parity error
82	11	Port checksum error
84	8	Port character error
91	9	Port address error
99	28	Timeout error during data transmission in ceLink format

Following are examples for EPROM type 2732. The highest address is 0FFF.

a. No address inputs (default values)
RAM begin address: 0000 RAM end address: 0FFF
EPROM begin address: 0000 EPROM end address: 0FFF

b. Additional input of block size 0A00 (command 0A00;)
RAM begin address: 0000 RAM end address: 09FF
EPROM begin address: 0000 EPROM end address: 09FF

c. Additional input of RAM begin address 1000 (command 1000<)
RAM begin address: 1000 RAM end address: 19FF
EPROM begin address: 0000 EPROM end address: 09FF

d. Additional input of EPROM begin address 0600 (command 0600:)
RAM begin address: 1000 RAM end address: 19FF
EPROM begin address: 0600 EPROM end address: 0FFF

e. The following additional inputs cause address errors:
0A01; EPROM end address: 1000 (exceeds EPROM maximum address)
0601: EPROM end address: 1000 (exceeds EPROM maximum address)
1601< RAM end address: 2000 (not available for 8K RAM)

CHAPTER 8

SERIAL PORT

8.1 INTRODUCTION

The RS-232C serial port, located at the top left corner of the front panel (see figure 1), is used to transfer data to/from the Model-804. In KEY-mode (Chapter 5) the serial port is used to input and output data between a remote station and the RAM of the Model-804 (functions 3 and 9),and to transmit data from the EPROM to a remote station (function F). In REM-mode (Chapter 7)the serial port is used to control the Model-804 from an external computer.

This chapter contains supplementary information on the RS-232C serial port. The first part of the chapter deals with the specific role of each pin on the RS-232C socket. The second section contains information on the initialization at the remote terminal, and complements Chapter 4, on the initialization of the Model-804. This chapter also contains information on error handling, and ends with a detailed section on the data formats used for data transfer.

8.2 CONNECTION DETAILS

External equipment with an RS-232C port can be connected to the Model-804 via the 25-pin socket marked RS-232C. The connection details are printed on the front panel. Below is an exact detailing of the functions of each pin.

Pin 1	EAR (Protective Earth) Connected directly to the main earth by a 1 Mohm resistor to the signal ground GND.
Pin 2	TxD (Transmit Data) This output signal carries the data sent from the Model-804 to the external device. In the transmission gaps this output is held at low level. For enabling the transmitter CTS and DSR inputs must be at HIGH level. By sending CTRL-S (CTRL-Q) to the Model-804 the transmitter will be disabled (enabled).
Pin 3	RxD (Received Data) This input carries the data to be received by the Model-804 from the external device. To activate the receiver DSR input must be at HIGH level.
Pin 4	RTS (Request To Send) Via this output the Model-804 indicates that it is ready to receive data, and that the external device can activate its

transmitter. For rates higher than 2400 baud it is important to use the RTS output connected to the CTS input of the external decvice. If the Model-804 receives more than one character when RTS is LOW, the extra characters cannot be processed and is lost. The error message OVERFLOW ERROR is displayed.

Pin 5 CTS (Clear To Send)
Via this input the external equipment indicates that it is ready to transmit data. This pin must be HIGH before the Model-804 will send data. The transmission is immediately stopped when this input goes LOW even during the transmission of character.

Pin 6 DSR (Data Set Ready)
Via this input the external equipment indicates its active state. It must be HIGH for every port operation. Compared to the CTS input the DSR does not break off the transmission of one character.

Pin 7 GND (Signal Ground)
Connected to protection earth by an 1 Mohm resistor.

Pin 9 +12 (HIGH Level)
This pin always has a high level. If either or both CTS and DSR are not used then they must be connected to pin 9.

Pin 15 Inverted RTS-signal (see pin 4)

Pin 20 DTR (Data Terminal Ready)
When the Model-804 is switched on this output is HIGH, except while pressing a MODE-key or in SIM-mode.

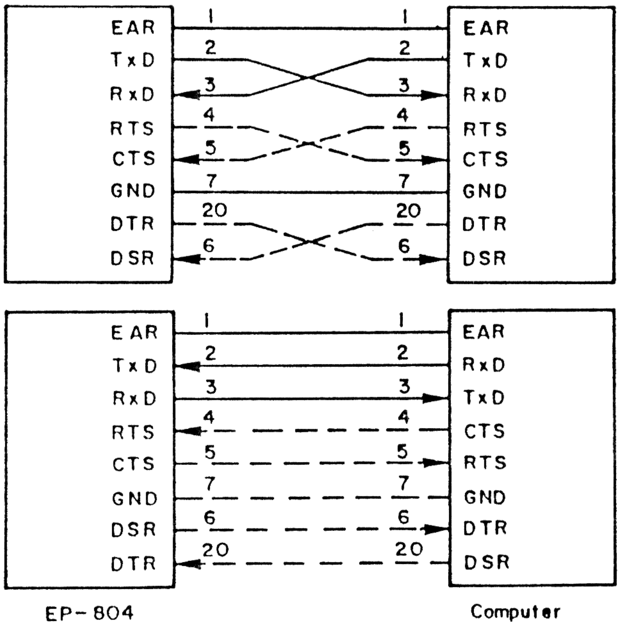


FIGURE-6 EXAMPLE OF CONNECTION TO THE SERIAL PORT

8.3 INITIALIZATION

Prior to using the serial port, the parameters at both the Model-804 and the external device must be set to corresponding values. The following section describes the different steps of initialization made in function 0. Refer to Chapter 4 for further details on initialization.

8.3.1 DATA FORMAT

There are several data formats which are described in detail in section 8.5. The format is selected by accepting one of the selected texts on display.

8.3.2 BAUD RATE

One of the values 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600 or 19200 baud is selected by accepting the appropriate display suggestion.

8.3.3. NUMBER OF STOPBITS

In the initialization function the choice is made between one or two stopbits. If during the input a false stopbit is found the error message FRAMING ERROR is displayed and must be acknowledged by pressing one of the MODE keys.

Level definition: startbit = space = zero databit : +3V...V...+15V
stopbit = mark = one databit : -3V...V...-15V

8.3.4 FRAME

In the initialization function the choice can be made between 7 or 8 data bit frame.

8.3.5 PARITY BIT

The choice is made between the following parity types:

- ODD PARITY and EVEN PARITY	Transmission of 7 data bits and the parity bit between startbit and stopbits. Parity is evaluated during input and if an error is encountered an error message will appear.
- SPACE PARITY	Transmission of 7 data bits and one space bit between startbit and stopbits. No evaluation during input.

- NONE PARITY	Transmission of only 7 data bits between startbit and stopbit.

If a false parity is found during input, the error message PARITY ERROR is displayed, which must be acknowledged by pressing one of the mode-keys.

8.3.6 ECHO MODE

In the initialization function the choice can be made between full echo on and echo off. In the echo on mode, data is transmitted simultaneously in both directions. Received input data is automatically returned (Auto Echo). In the echo off mode data can only be transmitted in one direction at a time.

8.3.7. ADDRESS ERROR HANDLING WITH DEVELOPMENT SYSTEM FORMATS

If the address information is to be transmitted via the port, ADDRESS FROM I/O is chosen in initialization. In this case only addresses in the range designated in RAM-begin and RAM-end addresses, are allowed.

If the parameter DISPLAY ADDR-ERR has been specified in initialization, addresses out of range cause an error message and the function to abort. If however, the parameter IGNORE ADD-ERR has been specified then all the concerned data are ignored. In either case only addresses that are in the defined RAM-begin address and RAM-end address are acceptable.

EXAMPLE:

A 12K byte program is to be loaded into 4K byte EPROMs. The loaded addresses are between 2000 and 4FFF. The Model-804 has a RAM capacity of 8K byte. The development system can only output the whole program. Therefore the splitting up of the program must be handled by the Model-804. First the combination ADDRESS FROM I/O and IGNORE ADD-ERR must be input. The programming is executed in two steps:

1. Base address = 2000
   Load data between RA = 2000 and RE = 3FFF.
   Program first and second EPROM.

2. Base address = 4000
   Load data between RA = 4000 and RE = 4FFF.
   Program third EPROM.

During the second step the data in the address range from 2000 to 3FFF are ignored. The Model-804 starts loading data from address 4000.

8.3.8 ADDRESS DISPLAY HANDLING DURING RS-232C FUNCTIONS

Usually the data RAM address is displayed during transmissions via the RS-232C interface, to control the load operation. For transmission in baud rates higher than 2400 baud, handshaking is needed. To increase the speed of transmission the display may be switched off. Then 4800 baud or 9600 baud can be used without handshaking (see also section 8.1). In REM-mode the RS-232C address display is always switched off (except in STANDARD remote all parameters are kept unchanged).

8.4 ERROR HANDLING

8.4.1 TRANSMISSION AND HARDWARE FAULTS

If an error in either parity framing or overflow occurs the data input is stopped. In KEY-mode the error message is displayed at the alphanumeric display (see chapter 5). In REM-mode the appropriate error status bits are set (see chapter 7). To reset the fault a mode key must be operated (hardware reset).

8.4.2 CHARACTER ERROR

If there is a character error in data input the error message CHARACTER ERROR is displayed. In manual input format, the data input can be repeated without changing the address, after acknowledgement with the ENTR-key. In all other formats the input is aborted. In REM mode the appropriate error status bits are set (see section 7). Data word errors (e.g. false block type in development system formats) are handled in the same way, however the function abort may be slightly delayed.

8.4.3 CHECKSUM AND ADDRESS ERRORS

If a checksum or address error occurs in development system formats the Model-804 reacts in the same way as in a data word error. In KEY-mode the message CHECKSUM ERROR or the relevant address error information (see section 5) is displayed. In REM-mode the applicable bits in the error status word are set.

8.5 DATA FORMATS

Data transfer between the Model-804 and external equipment via the serial port can be realized using one of six data formats. The desired data format is selected in the initialization function 0. See section 4.

The following is a detailed description of the data formats.

8.5.1 MANUAL

This format produces a legible record. The MANUAL format is used to input hexadecimal data from a terminal.

Data output:
Output format: 16 data bytes to the line.

EXAMPLE:

```
R0000   00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF (CR)
R0010   10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F (CR)
R0020   02 12 22 32 42 52 62 72 82 92 A2 B2 C2 D2 E2 F2 (CR)
```

Address of the following data byte
Data origin information (R = RAM, P = PROM)

Data input:

The Model-804 will transmit the address in the format described above. It will then recognize only hexadecimal characters, and will ignore all other characters.

A SPACE or a RETURN is used to either end a line or enter the data. RETURN, used as data entry, will start a new line beginning with the incremented address. The data finally stored in the RAM is composed of the two hex characters preceding the SPACE or RETURN character.

An invalid character appearing between the first HEX character and the enter character will generate the message CHARACTER ERROR, and the data byte to be processed will be set to zero. The RAM address will be incremented and the next HEX character awaited. An input of 16 bytes will cause the automatic start of a new line.

8.5.2 HEX-SPACE

In HEX format every data byte is given in the form of two ASCII-coded hexadecimal characters. SPACE (SPO, 20H) is the data entry character.

Data output:
Output format: 16 data bytes per line.

EXAMPLE:

```
00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF (CR)
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F (CR)
02 12 22 32 42 52 62 72 82 92 A2 B2 C2 D2 E2 F2 (CR)
```

Data input:

Until the data entry character 'SP' is recognized, all non-hexadecimal characters will be ignored. The data finally stored in RAM will be composed of the two hex characters preceding a space or carriage return.

If an invalid character or only one hex character is given between the first HEX character and enter, an error indication will be displayed and the affected data byte will be set to 0. Pressing the ENTR-key will abort the function.

8.5.3 ASCII-BNPF

In BNPF format every bit of a data byte is set with a 'N' (log.0 = LOW) or a 'P' (log.1 = HIGH). Each data byte starts with a 'B' followed by eight 'N' or 'P' characters. These represent the eight bits of the word, and end with an 'F'.

Data format:
Output format: 4 data bytes to a line.

EXAMPLE:

```
BNNNNNNNNF  BNNNPNNNPF  BNNPNNNPNF  BNNPPNNPPF (CR)
BNPNNNPNNF  BNPNPNPNPF  BNPPNNPPNF  BNPPPNPPPF (CR)
BPNNNPNNNF  BPNNPPNNPF  BPNPNPNPNF  BPNPPPNPPF (CR)
BPPNNPPNNF  BPPNPPPNPF  BPPPNPPPNF  BPPPPPPPPF (CR)
```

Data input:

The character 'B' must be entered to start the input, followed by 8 'N' or 'P' characters, and the character 'F'.

If an invalid character follows a 'B', the Model-804 will display a character error indication. The affected byte will be set to 0. The function is aborted with the ENTR-key.

8.5.4 INTEL-HEX

This hexadecimal ASCII-coded format is used in INTEL and several other development systems. The data format is as follows:

```
:WWAAAATTDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDSS
```

:	start character (3A H)
WW	number of data bytes of the following record (2 HEX characters)
AAAA	begin address of this record (4 HEX characters)
TT	record type (2 HEX characters): TT = 00 : data record TT = 01 : end record TT = 02 : address extension record TT = 03 : start address record.
DD	data bytes (2 HEX characters for each byte)
SS	checksum of the record (2 HEX characters): The sum of all the 8 bit bytes including WW, AAAA and TT.

Data output:

Output format: 16 data bytes per record (line).
Address extension and start address records are not generated.

EXAMPLE:

```
 WWAAAATT<----------------DD---------------->SS
| |   | |                                    | |
:1000000000112233445566778899AABBCCDDEEFFF8 (CR)
:10001000101112131415161718191A1B1C1D1E1F68 (CR)
:1000200002122232425262728292A2B2C2D2E2F230 (CR)
:00000001FF (CR)
                 ^
                 |
                 └─end record
```

Data input:

Input begins with the ':' character, followed by hex characters, and ended with the checksum. Any character that precedes ':' will be ignored. If an illegal character is entered, an error indication will be displayed, and the relevant data byte will be set to 0. Press ENTR to abort the function.

The first two hex characters (WW) indicate the data byte number for the following record. The last two characters (SS) must have the correct checksum. Otherwise a checksum error indication will be displayed.

Address extension record:

This record is usually used for INTEL-HEX 86/88.
The record length is always 02, the address AAAA is zero and the record type TT is 02. The following two data bytes, DDDD, are the bits 4 to 19 of the segment base address. The bits 0 to 3 are always 0. The addresses of the following data records are modified by adding this segment base address. The bits 0 to 15 of the resulting address are then taken by the Model-804 as loading addresses. If during transmission any changes in bits 16 to 19 occur the Model-804 will give an address error indication (ADDR ERR EXT). The segment base address will not be changed until a new extension record is received. Before receipt of the first address extension record the segment address is set to 0.

Data record:

The first two hex characters (WW) define the record length. When ADDRESS FROM I/O has been chosen in the initialization process, the address AAAA and the segment base address are used to calculate the load address of the first data word. If ADDRESS FROM KEY was initialized the data will be loaded starting at the RAM begin address. The type record is TT = 00. The data bytes in the remainder of the record will be processed sequentially by the Model-804.

Start address record:

The record length is always 04, the address AAAA is 0, and the record type TT is 03. Start address records are ignored by the Model-804.

End record:

The record length WW is always 00. The address AAAA is 0 and the record type is TT = 01. After receiving the end record the data input is terminated.
Note: In REM-mode the checksum (SS) of the end record will not be generated or processed.

8.5.5 MOTOROLA EXORCISER

This hexadecimal format in ASCII code is compatible to the data format used by the MOTOROLA EXORCISOR development systems. It is arranged as follows:

```
SNWWAAAADDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDSS
```

S	start characters ('S', 53H)
N	record type (one hex character)

N = 1 : data record with 16-bit address
N = 2 : data record with 24-bit address
N = 3 : data record with 32-bit address
N = 7 : end record after S3 records
N = 8 : end record after S2 records
N = 9 : end record after S1 records

WW	number of bytes (2 hex characters) of the following record, including address, data bytes and checksum.
AAAA (AAAA)	begin address of the following record (4 to 8 hex characters)
DD	Data bytes (2 hex characters)
SS	Checksum of the preceding record (2 hex characters). It is the one's complement of the sum of all 8-bit bytes including WW and AAAA (AAAA).

Data output:

The output is 16 data bytes per line.
Only S1- and S9-records are generated.

EXAMPLE:

```
SNWWAAAA<-----------DD------------>SS
||  |   |                          |
||  |   |                          |
S1130000001122334455667788 99AABBCCDDEEFFF4 (CR)
S113001010111213141516171819 1A1B1C1D1E1F64 (CR)
S113002002122232425262728292A2B2C2D2E2F22C (CR)
S9030000FC (CR)<-------+
                       |
                       +-------- end record
```

Data input:

The character 'S' begins data input. Any information prior to this character is ignored. The characters that follow must be a valid record type number otherwise a new record, beginning with the next start cha- racter, will be searched. Only hexadecimal characters are permissable, followed by the checksum. All erroneous characters generate a character error indication and the affected data byte is set to 0.

The final two hex characters (SS) of any record type are interpreted as checksum. A false checksum will set the data byte preceding the checksum to 0, a sum error message will be produced and the function will be aborted.

Data records:

Data records begin with S1, S2 or S3. The following two hex characters (WW) represent the number of data bytes of the next record. If the number of bytes is 3 in a S1 record (4 in a S2-record and 5 in a S3-record) the record contains no data and is ignored. Numbers smaller than this indicate a transmission error. The hex characters that follow determine the loading address of the first data byte of the record. There are four hex characters (16-bit address) in S1-records, 6 hex characters (24-bit address) in S2 records and 8 hex characters (32-bit address) in S3-records. Each address is filled internally with zeros to complete a 32-bit address. Any change in bits 16 to 31 during the transmission causes an address error message. If ADDRESS FROM I/O has been accepted in the initialization function, the bits 0 to 15 of the address are used as loading address. Otherwise (ADDRESS FROM KEY) the loading is performed starting at the RAM begin address (see section 8.2.6).

The following data bytes are processed sequentially by the Model-804 until the number of bytes WW is reached.

End records:

End records begin with S7, S8 or S9 and do not contain any data. The next 2 hex characters (WW) must be 03 for S9-records (04 for S8 records and 05 for S7 records). Otherwise a character error is displayed.

The following 4 (S9 record), 6 (S8 block) or 8 (S7 block) hex characters signify a start address and are ignored by the Model-804. Any end record will terminate the data input.

8.5.6 TEKTRONIX-MILLENNIUM

This hexadecimal, ASCII-coded format is used in Tektronix and Millennium development systems. The format is as follows:

```
/AAAAWWSSDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDPP
```

/	start character (slash, 2FH)
AAAA	start address of the following record (4 hex characters)
WW	number of data bytes in the following record. If WW = 00, it is an end record.
SS	checksum of AAAA and WW (2 hex characters). This is the 8-bit sum of all 4-bit nibbles.
DD	data bytes (2 hex characters each)
PP	checksum of all data bytes (2 hex charactrers). This is the 8-bit sum of all 4-bit nibbles.

Because the address is limited to 16 bits (4 hex characters) there is a special address extension record.

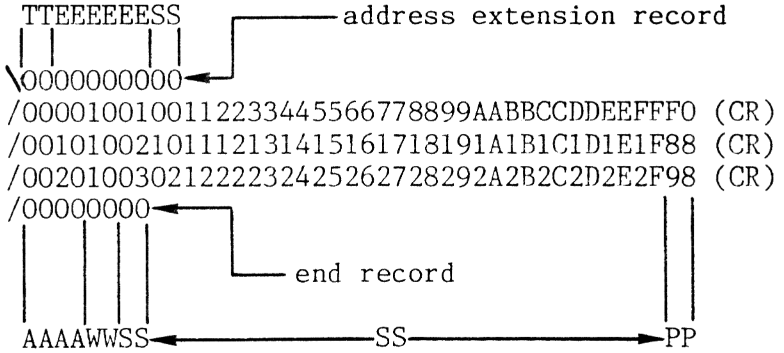
\TTEEEEEESS

\	start character (backslash, 5CH)
TT	record type (2 hex characters): always 00
EEEEEE	extension address (6 hex characters): This address will be added to the address of all following data records. The sum is the actual load address.
SS	checksum of TT and EEEEEE (2 hex characters). This is the 8-bit sum of all 4-bit nibbles.

Data output:

Every data record consists of 16 data bytes. Prior tq the data output an address extension record with the address 000000 is generated.

EXAMPLE:



```
\0000000000
/00001001001122334455667788899AABBCCDDEEFFF0 (CR)
/00101002101112131415161718191A1B1C1D1E1F88 (CR)
/00201003021222232425262728292A2B2C2D2E2F98 (CR)
/00000000
```

Data input:

The start character must precede any other data entry. All information entered before the start character is ignored. The start character may be a '/' (data or end block) or a 'ƒ' (address extension block). After the start character only hex characters are permissible until the record is finished. Other characters will cause an error message and the concerned data byte will be set to zero.

Data block:

The first four hex characters (AAAA) added to the address extension address are the load address of the first data byte in the record. If ADDRESS FROM I/O has been selected in the initialization function this address becomes the load address. The two hex characters that follow, represent the number of data bytes. The next two characters (SS) are the checksum of the preceding part of the record (i.e. AAAA and WW). They are followed by the data bytes which will be processed sequentially by the Model-804 until the number of bytes in the record is reached. The record is terminated with the checksum of all data bytes.

End record:

An end record is defined as a data record with the number of data bytes set to zero. The address is not used but the checksum must be correct.

Address extension record:

The first two hex characters (TT) after the special start character (' ') must be 00, otherwise a new record is searched. The following 6 characters are interpreted as the extension address. This address is added to all addresses of the following data records until the extension address is renewed. Bits 0 to 15 of the address sum are used as the load address by the Model-804. Bits 16 to 19 must not change during transmission, otherwise an address error indication (ADDRESS ERR EXT) is generated. The last two characters are the checksum of the record.

8.5.7 MOSTECH STANDARD Z80 OBJECT OUTPUT DEFINITION

Each record of an object module begins with a delimiter (colon or dollar sign) and ends with carriage return and line feed. A colon (:) is used for data records and end of file records. A dollar sign ($) is used for records containing relocation information and linking information. An Intel loader will ignore such information and allow loading of non-relocatable, non-linkable programs. All information is in ASCII.

Each record is identified as a type. The type appears in the 8th and 9th bytes of the record and can take the following values:

00 - data record
01 - end-of-file
02 - internal symbol
03 - external symbol
04 - relocation information
05 - module definition

Data Record Format:

Byte 1	Colon (:) delimiter
2-3	Number of binary bytes of data in this record. The maximum is 32 binary bytes (64 ASCII bytes).
4-5	Most significant byte of the start address of data.
6-7	Least significant byte of start address of data.
8-9	ASCII zeros. This is the "record type" for data.
10-	Data bytes.

Last two bytes - Checksum of all bytes except the delimiter, carriage return, and line feed. The checksum is the negative of the binary sum off all bytes in the record.

CRLF Carriage return, line feed.

END-OF-FILE RECORD (TYPE 0):

Byte 1	Colon (:) delimiter.
2-3	ASCII zeros.
4-5	Most significant byte of the transfer address of the program. This transfer address appears as an argument in the 'END' pseudo-op of a program. It represents the starting execution address of the program.
6-7	Least significant byte of the transfer address.
8-9	Record type 01.
10-11	Checksum
CRLF	Carriage return, line feed.

CHAPTER 9

EPROM SIMULATION

9.1 INTRODUCTION

The third mode of operation of the Model-804 is SIM-mode. In SIM-mode the program that is loaded into user RAM can be tested, as though it were a programmed EPROM.

Locate the SIM socket at the lower left side of the front panel (refer to figure 1). This socket is used to connect the Model-804 to the equipment to be tested (microprocessor development board). Special adapter cables are available for all EPROM types. Refer to Appendix A for the order numbers.

The simulator facility offers the option of checking the unit under test including all I/O peripherals. All the Model-804 editing facilities can be used to debug the program. When the program is perfect the EPROM can be programmed and installed in the equipment.

SIM-mode is useful for:

- The Model-804 can be used as a development system for microprocessor development with short programs. Long programs, however, will require an assembler.
- Program development for simple control, test equipment, etc.
- Program changes (fitting into a service environment).

The EPROM type can be selected in either KEY-mode with function 0 (see chapter 4) or in REM-mode with command 'n' (see chapter 7). The capacity of the simulator address range is set automatically based on the EPROM selected.

9.2 TECHNICAL DATA

Capacity:	0.5, 1, 2, 4, 8, 16, 32 to 64K bytes, limited by the the RAM capacity of the Model-804.
RAM begin address of the simulator:	Always 0
RAM end address of the simulator:	According to the EPROM type
Address access time (TACC):	Max. 300ns (typ. 200ns)

Output access time (TEN)	CS0, CS1: max. 100ns (typ. 60ns) CE3, CE4: max. 140ns (typ. 100ns)
Output switch-off time (TDIS)	CS0, CS1: max. 100ns (typ. 60ns) CE3, CE4: max. 140ns (typ. 100ns)
TTL inputs:	one LS-TTL-load without pull-up except ECE: one LS-TTL-load with 1kohm-Pull-Down
Tri-state outputs:	LOW-level: VOL < 0,4V IOL < 2,2mA HIGH-level: VOH > 2,4V -IOH < 2,2mA
Socket connection:	37-pin D-socket

9.3 TIMING CONSIDERATIONS

The following are the conditions required for outputting data to the SIM connector.

- Simulation started
- Inputs CS0 and CS1 must be LOW
- If ECE-input is HIGH (input open = LOW) then the following additional conditions must be met: CE3 = CE4 = LOW

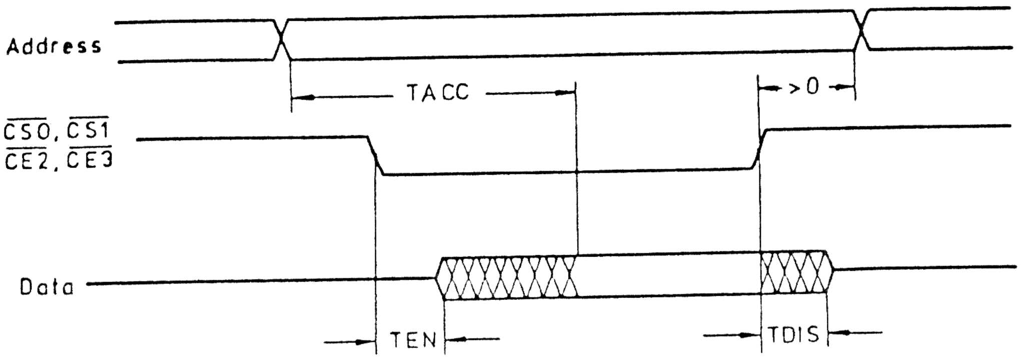


FIGURE 8 TIMING DIAGRAM

9.4 CONNECTION DETAILS

Figure 7 is a diagram of the SIM socket. The signal descriptions are printed on the front panel. Numbers correspond to the socket marking.

A0...A14	address input
D0...D7	data output
CS0, CS1	enable input
CE3, CE4	enable input
ECE	enable input for CE2 and CE3
NC	No connection
GND	Signal ground connected to protection earth by 1Mohm resistor.

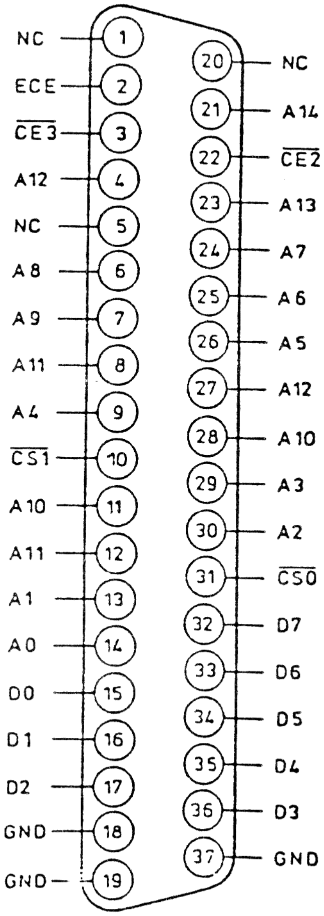


FIGURE 9 SIM CONNECTOR

9.5 OPERATION

Connect the proper simulator cable from the MODEL-804 to the test equipment. Before the simulator is started all the data inputs are high impedance.

When loading the simulation data into the RAM of the MODEL-804 the simulation address range will always start at zero, and ends at the highest address of the EPROM in use.

EXAMPLE:

The simulation range of the EPROM 2716 is 0...7FF.

The RAM size of the programmer must be of at least the EPROM capacity. All unused high address inputs do not have to be connected.

The simulation is started by pressing the SIM-key which sets the internal microprocessor to an idle state. The message

SIMULATION

is then displayed. The external equipment can then access the RAM data directly. Simulation is terminated by pressing either the KEY or REM mode keys, or the OFF-keys.

APPENDIX A

A TABLE OF PROGRAMMABLE DEVICES

CAPACITY IN BYTES	EPROM PART NO.	MODEL-804 DESCRIPTION	DEVICE CODE	SIM-CABLE ORDER NO.	ADAPTER NO.
ADVANCED MICRO DEVICES (AMD)					
1024	Am2708	TMS 2708	0084	2024	
1024	Am9708	TMS 2708	0084	2024	
2048	Am2716	2716	0000	2020	
2048	Am9716	2716	0000	2020	
4096	Am2732	2732	0010	2020	
4096	Am2732A	2732A	0011	2020	
4096	Am8751H	8751H	1551		8043
8192	Am2764	INT 2764	00B0	2021	
8192	Am2764A	AMD 2764A	00A1	2021	
8192	Am9864	I 2864	0180	2020	8046
8192	Am9761H	AM 8761H	1561		8043
16384	Am27128	INT 27128	00B4	2021	
16384	Am27128A	AMD 27128A	00A5	2021	
32768	Am27256	AMD 27256	00A8	2021	
65536	Am27512	AMD 27512	00AC	2021	
ELECTRONIC ARRAYS					
512	2704	TMS 2704	0080	2024	
1024	2708	TMS 2708	0084	2024	
2048	2716	2716	0000	2020	
EXEL					
2048	XL2816A	X 2816A	0143	2020	8046
2048	XL2817A	I 2817A	0162	2021	8046
2048	XL46C16	XL 46C16	01C0		
4096	XL46C32	XL 46C32	01C4		
8192	XL2864A	X 2864	0182	2021	8046
8192	XL2865A	I 2864	0180	2021	8046

CAPACITY IN BYTES	EPROM PART NO.	MODEL-804 DESCRIPTION	DEVICE CODE	SIM-CABLE ORDER NO.	ADAPTER NO.
FAIRCHILD SEMICONDUCTOR					
1024	2708	TMS 2708	0084	2024	
2048	2716	2716	0000	2020	
FUJITSU					
1024	MBM8518	TMS 2708	0084	2024	
1024	MBM2708	TMS 2708	0084	2024	
2048	MBM8516	2716	0000	2020	
2048	MBM2716	2716	0000	2020	
4096	MBM2732	2732	0010	2020	
4096	MBM2732A	2732A	0011	2020	
4096	MBM27C32	27C32	0044	2020	
4096	MBM27C32A	27C32A	0046	2020	
8192	MBM2764	INT 2764	00B0	2021	
8192	MBM27C64	NEC 27C64	00C0	2021	
16384	MBM27128	INT 27128	00B4	2021	
16384	MBM27C128	NEC 27C128	00C4	2021	
32768	MBM27256	INT 27256	00B8	2021	
32768	MBM27C256	NEC 27C256	00C8	2021	
65536	MBM27C512	INT 27C512	00CC	2021	
HITACHI					
2048	HN462716	2716	0000	2020	
2048	HN48016	HN 48016	0120	2020	
4096	HN462532	TMS 2532	0070	2026	
4096	HN462732	2732	0010	2020	
4096	HN482732A	2732A	0011	2020	
8192	HN482764	INT 2764	00B0	2021	
8192	HN27C64	NEC 27C64	00C0	2021	
8192	HN58064	HN 58064	0189	2021	8046
16384	HN4827128	INT 27128	00B4	2021	
32768	HN27256	INT 27256	00B8	2021	

CAPACITY IN BYTES	EPROM PART NO.	MODEL-804 DESCRIPTION	DEVICE CODE	SIM-CABLE ORDER NO.	ADAPTER NO.
INTEL					
512	2704	TMS 2704	0080	2024	
1024	2708	TMS 2708	0084	2024	
1024	8708	TMS 2708	0084	2024	
1024	8741A	8741A	1541		8045
1024	8748	8748	1548		8045
1024	8748H	8748H	1547		8045
1024	2758	2758	0020	2020	
1024	2758S1865	2716	0000	2020	
(only address range from 400...7FF can be used)					
2048	2716	2716	0000	2020	
2048	2815	I 2815	0140	2020	8041
2048	2816	I 2816	0141	2020	8041
2048	2816A	I 2816	0141	2020	8041
2048	2817	I 2817	0160		8044
2078	2817A	I 2817A	0162	2021	8046
2048	8742	8742	1542		8045
2048	8749H	8749H	1549		8045
2048	8755A	8755A	1555		8042
4096	2732	2732	0010	2020	
4096	2732A	2732A	0011	2020	
4096	8744H	8751H	1551		8043
4096	8751	8751	1550		8043
4096	8751H	8751H	1551		8043
INTEL					
8192	2764	INT 2764	00B0	2021	
8192	2764A	INT 2764A	00B1	2021	
8192	27C64	INT 27C64	00C1	2021	
8192	87C64	INT 87C64	00C2		
8192	2864B	I 2864	0180	2021	8046
16384	27128	INT 27128	00B4	2021	
16384	27128A	INT 27128A	00B5	2021	
32768	27256	INT 27256	00B8	2021	
32768	27C256	INT 27C256	00C9	2021	
32768	87C256	INT 87C256	00CA		
65536	27512	INT 27512	00BC	2021	
65536	27513	INT 27513	00BF		
LATTICE					
8192	EE64K8	I 2864	0180	2021	8046

CAPACITY IN BYTES	EPROM PART NO.	MODEL-804 DESCRIPTION	DEVICE CODE	SIM-CABLE ORDER NO.	ADAPTER NO.
MITSUBISHI					
1024	M58732S	TMS 2708	0084	2024	
1024	M5L2708	TMS 2708	0084	2024	
2048	M5L2716	2716	0000	2020	
4096	M5L2732	2732	0010	2020	
8192	M5L2764	2764	0030	2021	
16384	M5L27128	INT 27128	00B4	2021	
MOSTEK					
1024	MK2708	TMS 2708	0084	2024	
2048	MK2716	2716	0000	2020	
MOTOROLA					
1024	MCM68708	TMS 2708	0084	2024	
1024	MCM68A708	TMS 2708	0084	2024	
2048	MCM2716	2716	0000	2020	
2048	MCM27A16	2716	0000	2020	
2048	TMS2716	TMS 2716	0088	2025	
2048	MCM27A16	TMS 2716	0088	2025	
4096	MCM2532	TMS 2532	0070	2026	
4096	MCM25A32	TMS 2532	0070	2026	
4096	MCM68732-0	MCM68732	0050	2026	
4096	MCM68732-1	MCM68764	0054	2026	
(however an address range from 1000...1FFF is to be used!)					
8192	MCM68764	MCM68764	0054	2026	
8192	MCM68766	MCM68766	0055	2026	

CAPACITY IN BYTES	EPROM PART NO.	MODEL-804 DESCRIPTION	DEVICE CODE	SIM-CABLE ORDER NO.	ADAPTER NO.
NATIONAL					
512	MN2704	TMS 2704	0080	2024	
1024	MM2708	TMS 2708	0084	2024	
1024	MM2758A	2758	0020	2020	
1024	MM2758B	2716	0000	2020	
(however an address range from 400...7FF is to be used!)					
2048	MM2716	2716	0000	2020	
2048	NMC27C16	27C16	0040	2020	
2048	NMC27C16H	27C16H	0041	2020	
2048	NSC9817	I 2817A	0162	2021	8046
4096	NMC2532	TMS 2532	0070	2026	
4096	NMC2732	2732	0010	2020	
4096	NMC27C32	27C32	0044	2020	
4096	NMC27C32H	27C32H	0045	2020	
8192	NMC2564	TMS 2564	0074	2028	
32768	NMC27C256	INT 27C256	00C9	2021	
NIPPON ELECTRIC (NEC)					
1024	uPD8741A	8741A	1541		8045
1024	uPD8748	8748	1548		8045
1024	uPD8748H	8748H	1547		8045
2048	uPD2716	2716	0000	2020	
2048	uPD8749H	8749H	1549		8045
2048	uPD8755A	8755A	1555		
4096	uPD2732	2732	0010	2020	
4096	uPD2732A	2732A	0011	2020	
8192	uPD2764	INT 2764	00B0	2021	
8192	uPD27C64	NEC 27C64	00C0	2021	
16384	uPD27128	INT 27128	00B4	2021	
32768	uPD27256	NEC 27256	00B9	2021	
32768	uPD27C256	NEC 27C256	00C8	2021	
32768	uPD27C256A	INT 27C256	00C9	2021	

CAPACITY IN BYTES	EPROM PART NO.	MODEL-804 DESCRIPTION	DEVICE CODE	SIM-CABLE ORDER NO.	ADAPTER NO.
OKI ELECTRIC					
1024	MSM2708	TMS 2708	0084	2024	
1024	MSM2758	2758	0020	2020	
2048	MSM2716	2716	0000	2020	
4096	MSM2732	2732	0010	2020	
4096	MSM2532	TMS 2532	0070	2026	
8192	MSM2764	2764	0030	2021	
ROCKWELL					
2048	R2816A	X 2816A	0143	2021	8046
2048	RS2816A	X 2816A	0143	2021	8046
2048	R5213	I 2816A	0142	2020	8046
2048	R5213H	I 2816A	0142	2020	8046
2048	R5516A	X 2816A	0143	2020	8046
4096	R87C32	27C32A	0046	2020	
8192	R27C64P	27C64	0048	2021	
8192	R52B33	52B33	0188	2021	8046
8192	R52B33H	52B33	0188	2021	8046
8192	R87C64	27C64	0048	2021	
SEEQ					
2048	2816A	X 2816A	0143	2020	8046
2048	2816AH	X 2816A	0143	2020	8046
2048	2817A	I 2817A	0162	2021	8046
2048	2817AH	I 2817A	0162	2021	8046
2048	52B13	I 2816A	0142	2020	8046
2048	52B13H	I 2816A	0142	2020	8046
2048	5516A	X 2816A	0143	2020	8046
2048	5517A	I 2817A	0162	2021	8046
2048	5517AH	I 2817A	0162	2021	8046
8192	2764(5133)	INT 2764	00B0	2021	
8192	52B33	52B33	0188	2021	8046
8192	52B33H	52B33	0188	2021	8046
16384	27128(5143)	INT 27128	00B4	2021	
32768	27C256	INT 27C256	00C9	2021	

CAPACITY IN BYTES	EPROM PART NO.	MODEL-804 DESCRIPTION	DEVICE CODE	SIM-CABLE ORDER NO.	ADAPTER NO.
SGS-ATES					
2048	M2716	2716	0000	2020	
4096	M2532	TMS 2532	0070	2026	
THOMSON					
2048	ET2716	2716	0000	2020	
2048	ETC2716	27C16	0040	2020	
4096	ETC2732	27C32	0044	2020	
8192	ET2764	INT 2764	00B0	2021	
16384	ET27128	INT 27128	00B4	2021	
TEXAS INSTRUMENTS					
512	TMS2704	TMS 2704	0080	2024	
1024	TMS2708	TMS 2708	0084	2024	
1024	TMS27L08	TMS 2708	0084	2024	
2048	TMS2516	2716	0000	2020	
2048	TMS2716	TMS 2716	0088	2025	
4096	TMS2532	TMS 2532	0070	2026	
4096	TMS2732A	2732A	0011	2020	
8192	TMS2564	TMS 2564	0074	2028	
8192	TMS2764	INT 2764	00B0	2021	
16384	TMS27128	INT 27128	00B4	2021	
16384	TMS27C128	INT 27C128	00C5	2021	
32768	TMS27256	INT 27256	00B8	2021	
32768	TMS27C256	INT 27C256	00C9	2021	

CAPACITY IN BYTES	EPROM PART NO.	MODEL-804 DESCRIPTION	DEVICE CODE	SIM-CABLE ORDER NO.	ADAPTER NO.
TOSHIBA					
512	TMM321	TMS 2704	0080	2024	
1024	TMM322	TMS 2708	0084	2024	
2048	TMM323	2716	0000	2020	
4096	TMM2732D	2732	0010	2020	
8192	TMM2764D	2764	0030	2021	
16384	TMM27128D	27128	0034	2021	
32768	TMM27256D	NEC 27256	00B9	2021	
32768	TC57256D	NEC 27C256	00C8	2021	
VLSI TECHNOLOGY (VTI)					
32768	VT27C256	INT 27C256	00C9	2021	
XICOR					
512	X2804A	X 2804A	0158	2020	8046
2048	X2816A	X 2816A	0143	2020	8046
2048	X2816B	X 2816A	0143	2020	8046
2048	X2816H	X 2816A	0143	2020	8046
8192	X2864A	X 2864	0182	2021	8046
8192	X2864B	X 2864	0182	2021	8046
8192	X2864H	X 2864	0182	2021	8046
8192	X28C64	X 2864	0182	2021	8046
32768	X28256	X 28256	0192	2021	8046
32768	X28C256	X 28256	0192	2021	8046

APPENDIX B

MODIFICATION INSTRUCTIONS FOR UPGRADING THE MODEL-804 SOFTWARE

DIGELEC will be continually updating the software for the Model-804. These system updates will be available in revision packages, consisting of replacement EPROMs. The user can upgrade his own programmer by following the directions below.

EQUIPMENT NEEDED TO UPGRADE THE MODEL-804

The Model-804 is upgraded by replacing EPROMs on the PC board. This is a simple process, requiring minimal skill and equipment.

Equipment required:

1. The upgrading kit containing replacement EPROMs:
2. A screwdriver

PREPARATIONS FOR UPGRADING

WARNING:

UNPLUG YOUR MODEL-804 BEFORE OPENING THE CASE

1. Lower the lever on the EPROM socket so that it lies flush with the front panel. Observe this precaution to avoid damaging the lever.
2. Turn the Model-804 over, so that the front panel is facing down on the table. We suggest that you cover the surface of your table, with a soft material, to protect the front panel against scratches.
3. Unscrew the four screws at the corners of the Model-804 base.
4. Lift the base and place it, face down, adjacent to the top section.

Your Model-804 is now open before you. The main PC board, labeled 804-0197B sits inside the front panel, and is connected to the board on the inside of the Model-804 back by a flat cable.

Turn the unit so that the main PC board is facing you, and the open cover is above it. This will facilitate easy access to the relevant EPROMs.

UPGRADING THE MODEL-804

1. Locate the EPROMs D41 and/or D42 labeled

804-2 REV XX with checksum SS
804-3 REV XX with checksum SS.

at the wall of the main PC board closest to you.

(XX= last software revision, SS= checksum of the EPROM)

2. Remove these two EPROMs (above) using the screwdriver.

3. Insert the new EPROMs.

Note: If your upgrading kit contains only one EPROM, insert it in the location marked on its label.

CLOSING THE CASE AND SYSTEM TEST

1. Replace the back of the Model-804, and with the programmer lying face down on the table, replace the screws.

2. Turn the programmer over, and plug the system into the main power.

```
POWER FAILURE
```

will appear on the display. Press ENTR . The message

```
SELF CHECK
```

will be displayed while the programmer performs a system check. When completed the message

```
TEST OK   1FFF   XX
```

will be displayed, indicating that the system is in perfect working condition. The highest address is 1FFF (8k bytes) and the system software has been upgraded to revision XX.

APPENDIX C

INTERFACING THE MODEL-804 WITH THE INTEL MDS

SCOPE

The DIGELEC Model-804 Programmer may be interfaced with the INTEL MDS when operating in two different modes: the KEY Mode and the REM Mode.

In the KEY Mode, the data block parameters are defined directly from the Model-804 keyboard. The interface software for the KEY Mode procedures is located on the DIGUTIL diskette.

In the REM Mode (Remote Control), you initialize the communications operation from the Model-804 keyboard, and then specify the data block parameters via the MDS. The Model-804 REM Mode software is compatible with the DATA I/O interface program. This program permits MDS remote control and editing of all Model-804 operations. The interface program is available from the INTEL User's Library (INSITE).

Communication between the Model-804 and the MDS is made by connecting the RS-232C port on the Model-804 and serial channel 2 (J3) on the MDS.

This Application Note describes the procedures for data transmission setup and operation in the KEY and REM Modes.

SECTION 1 : DATA TRANSMISSION IN THE MODEL-804 KEY MODE

1.1 REQUIRED EQUIPMENT

INTEL:	Series II or Series III Microcomputer Development System (MDS) with ISIS-II.
DIGELEC:	Model-804 EE/EPROM Programmer; DIGUTIL diskette, version 1 or 2.

1.2 HARDWARE INTERFACE

Set the INTEL interface jumpers on the IOC board as follows:

W1	A-B , C-D
W2	A-B
W7	A-B , C-D

Wire the RS-232C interface cable as follows:

INTEL MDS (J3)		DIGELEC MODEL-804	
SIGNAL	PIN NO.	PIN NO.	SIGNAL
TXD	2	3	RXD
RXD	3	2	TXD
RTS	4	5	CTS
CTS	5	4	RTS
DSR	6	20	DTR
DTR	20	6	DSR
GND	7	7	GND

1.3 DATA FORMAT TRANSLATION

When operating in the KEY Mode, data may be transmitted only in the Hex format. Therefore, any file initially built in the Object format must be converted into the Hex form. This may be accomplished on the MDS using the OBJHEX utility program supplied with ISIS-II.

Type in the following command to translate a file from the Object format to the Hex format:

```
OBJHEX FILENAME.OBJ TO FILENAME.HEX
```

where FILENAME.OBJ is the Object file to be changed;
FILENAME.HEX is the INTEL Hex format of that file.

Press carriage return.

1.4 INITIALIZATION PROCEDURES

1.4.1 MODEL-804 INITIALIZATION FOR TRANSMISSION

With the Model-804 activated, and the Self-check completed, press the ENTR key. The prompt

'SELECT FUNCTION'

will be displayed.

Press '0' on the Model-804 hexadecimal keyboard. Verify that the LED adjacent to the function labelled '0 INITIALIZE' on the front panel is lit.

Locate the appropriate option for the parameters listed below by scrolling with the NEXT key. Acknowledge each option by pressing the ENTR key.

Parameter	Option
Data Format	FORMAT HEX-INTEL
Baud Rate	BAUDRATE 4800
Stopbit	ONE STOPBIT
Parity	SPACE PARITY
Duplex	HALF DUPLEX
Address Source	ADDRESS FROM I/O
Address Error	IGNORE ADD-ERR

1.4.2 MDS INITIALIZATION FOR TRANSMISSION

With the ISIS-II loaded, insert the DIGUTIL diskette into one of the MDS drives. Type in the command

```
BAUDS2
```

and press carriage return.

You will be prompted to select the desired channel and the baud rate. Enter '2' for channel 2, then press carriage return.

Enter '6' for 4800 baud, then press carriage return.

To exit out of the initialization mode, enter '0'. Press carriage return.

1.5 DATA TRANSFER PROCEDURES

1.5.1 TRANSMITTING DATA FROM THE MODEL-804 TO THE MDS

On the MDS, type in the following command:

```
FR8032 BUFADD BLOCKLEN FILENAME.HEX
```

where FR8032	is the program for transmitting data from the Model-804.
BUFADD	is the offset address and must be 4200H or greater.
BLOCKLEN	is FFFF.
FILENAME.HEX	is the MDS Hex formatted file where the data will be stored.

Press carriage return.

On the Model-804, press key 9 of the hexadecimal keyboard. Key 9 corresponds to the function 'OUT:RS-232C'.

The prompt

'RAM BEGIN ADDRESS 0'

will be displayed. Enter in the first address of the data block to be transferred. Press the ENTR key.

The next prompt

'RAM END XXXX'

will be displayed. Enter the last RAM address (XXXX) to be transferred, then press ENTR. This initiates transmission of data from the Model-804 to the MDS via the serial port.

1.5.2 DATA TRANSMISSION FROM THE MDS TO THE MODEL-804

Verify that the Model-804 has sufficient memory for insertion of new data. (The last RAM address is displayed at the end of the Self-check.)

Press key 3 on the Model-804 hexadecimal keyboard. Key 3 corresponds to the function 'IN:RS-232C'.

The prompt

'RAM BEGIN 0'

will be displayed. Enter the first RAM address of the data block to be transferred. Press the ENTR key.

The next prompt

'RAM END XXXX'

will be displayed. Enter the last RAM address (XXXX) to be transferred, then press the ENTR key.

On the MDS, type in the following command:

```
T08032 BUFADD BLOCKLEN FILENAME.HEX
```

where T08032	is the program for transferring data from the MDS.
BUFADD	is the offset address and must be 4200H or greater.
BLOCKLEN	is the number of bytes to be transferred; block length should not exceed 0BDDF for a 64K MDS or 3DFFH for a 32K MDS.
FILENAME.HEX	is the file to be transferred in INTEL Hex format.

Press carriage return. Data transmission will be initiated.

Upon completion of data transfer, the Model-804 will display the last address it received.

SECTION 2 : DATA TRANSMISSION IN THE MODEL-804 REM MODE

2.1 REQUIRED EQUIPMENT

INTEL:	Series II or Series III Microcomputer Development System with ISIS-II.
DIGELEC:	Model-804 EE/EPROM Programmer; DATA I/O interface program (available from the INTEL INSITE Library).

2.2 HARDWARE INTERFACE

Set the INTEL interface jumpers on the IOC board as follows:

W1	A-B , C-D
W2	A-B
W7	A-B , C-D

Wire the RS-232C interface cable as follows:

INTEL MDS (J3)			DIGELEC MODEL-804	
SIGNAL	PIN NO.		PIN NO.	SIGNAL
TXD	2	------------	3	RXD
RXD	3	------------	2	TXD
RTS	4		5	CTS
CTS	5		6	DSR
			9	12V
GND	7	------------	7	GND

NOTE

Pins 4 and 5 must be jumpered at the MDS end, and pins 5,6, and 9 should be jumpered at the Model-804 end.

2.3 DATA FORMAT TRANSLATION

When operating in the REM Mode, data may be transmitted only in the normal Object format. Therefore, any file initially built in the Hex format must be converted into the Object format. This may be accomplished on the MDS using the HEXOBJ utility program supplied with ISIS-II.

Type in the following command to translate a file from the Hex format to the Object format:

```
HEXOBJ FILENAME.HEX TO FILENAME.OBJ
```

where FILENAME.HEX is the Hex file to be changed;
FILENAME.OBJ is the INTEL OBJ format of that file.

Press carriage return.

2.4 DATA TRANSFER PROCEDURES

2.4.1 MODEL-804 SETUP FOR DATA TRANSMISSION

With the Model-804 activated, press the REM Mode key. The default

'STANDARD REMOTE'

will be displayed.

Using the NEXT key, scroll through the different remote options until

'INTEL MDS REMOTE'

is displayed.

Acknowledge this option by pressing the ENTR key. The Model-804 is now ready to receive commands from the MDS.

2.5.2 MDS SETUP FOR DATA TRANSMISSION

On the MDS, type in the command

```
DATAIO
```

and press carriage return. The interface program will be called up. This program includes a help function to aid you in selecting the appropriate transmission operation.

Call up the HELP menu by typing in the command

```
HELP
```

and press carriage return. Select the appropriate operation from the HELP menu and specify a data block. A few of the HELP operations are described below:

```
LOAD FILENAME XXXX TO YYYY INTO ZZZZ (cr)
```

Load locations XXXX to YYYY of the OBJ file called 'FILENAME' into the MDS RAM beginning at location ZZZZ.

```
PROGRAM XXXX TO YYYY START ZZZZ (cr)
```

Program locations XXXX to YYYY of the MDS RAM into the EE/EPROM in the Model-804, beginning at location ZZZZ.

```
READ XXXX TO YYYY INTO ZZZZ (cr)
```

Load location XXXX to YYYY of the EE/EPROM into the MDS RAM beginning at location ZZZZ.

APPENDIX D

OPERATING AT THE DEVELOPMENT SYSTEM PMDS-I

The software of the REM-mode is compatible with the programming software of the PHILIPS PMDS-development system.

Using the Model-804 with the PMDS-I system requires an interface card TSEC (Triple Serial Interface Unit, PM 4473). It is connected at X2 (Channel 2, PROM programmer port). See Chapter 2 for information on ordering the corresponding interface cable.

Before switching on the following jumpers at the TSEC-card must be localized, and if necessary, adjusted according to the PMDS user manual.

Jumper X6: Pin 1---Pin 2
Jumper X7: Pin 1---Pin 2
Jumper X19: Pin 2---Pin 3

Select the remote type 'PMDS1 REMOTE' which sets serial port parameters as follows:

- Offset address	0
- Data format	INTEL-HEX
- Baud rate	9600
- Number of stopbits	1
- Parity type	odd
- Duplex type	half duplex
- ADDR FROM I/O	
- DISPLAY ADD ERR	
- RS-232C DISPL OFF	

Before activating the PROM processor at the PMDS-system select the required EPROM type in KEY-mode (see Chapter 4). Then press the REM-key at the Model-804. All functions described in the PMDS-system operating manual are available. To change the EPROM type, the PROM program at the PMDS must be aborted and started anew. Otherwise the PMDS system will not recognize a new EPROM type.

The address range is fixed from 0 to 7FF (2k byte). EPROMs with more capacity must be operated block by block, setting the EPROM begin address (0, 800, 1000, 1800,...) in the PMDS commands.

OPERATING AT THE DEVELOPMENT SYSTEM PMDS-II

The software of the REM mode is compatible with the programming software of the PHILIPS PMDS-development system.

Using the Model-804 with the PMDS-II system requires an interface card TSEC (Triple Serial Interface unit, PM4473). It is connected at X3 (Channel 3). See Chapter 2 for information on ordering the the appropriate interface cable.

Before switching on the following jumpers at the TSEC card must be localized, and when necessary adjusted according to the PMDS user manual.

Jumper X4:	Pin 3---Pin 6	Jumper X11:	Pin 1---Pin 2
Jumper X4:	Pin 4---Pin 7	Jumper X12:	Pin 1---Pin 2
Jumper X4:	Pin 12--Pin 15	Jumper X13:	Pin 2---Pin 3
Jumper X4:	Pin 13--Pin 16	Jumper X14:	Pin 2---Pin 3
Jumper X5:	Pin 1---Pin 2	Jumper X15:	Pin 1---Pin 2
Jumper X5:	Pin 3---Pin 4	Jumper X16:	Position 16
Jumper X8:	Pin 1---Pin 2	Jumper X17:	Position 13
Jumper X9:	Pin 1---Pin 2	Jumper X18:	Position 12
Jumper X10:	Pin 1---Pin 2	Jumper X20:	Pin 2---Pin 3

Select the remote type 'PMDS1 REMOTE' which sets serial port parameters as follows:

- Offset address	0
- Data format	INTEL-HEX
- Baud rate	4800
- Number of stopbits	1
- Parity type	odd
- Duplex type	half duplex
- ADDR FROM I/O	
- DISPLAY ADD ERR	
- RS-232C DISPL OFF	

Before activating the PROM processor at the PMDS-system select the required EPROM type in KEY-mode (see Chapter 4). Then press the REM-key at the Model-804. All functions described in the PMDS-system operating manual are available. To change the EPROM type, the PROM program at the PMDS must be aborted and started anew. Otherwise the PMDS system will not recognize a new EPROM type.

The address range is fixed from 0 to 7FF (2k byte). EPROMs with more capacity must be operated block by block, setting the EPROM begin address (0, 800, 1000, 1800,...) in the PMDS commands.